scientific reports

Check for updates

OPEN

A fuzzy logic-based secure hierarchical routing scheme using firefly algorithm in Internet of Things for healthcare

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

The Internet of Things (IoT) is a universal network to supervise the physical world through sensors installed on different devices. The network can improve many areas, including healthcare because IoT technology has the potential to reduce pressure caused by aging and chronic diseases on healthcare systems. For this reason, researchers attempt to solve the challenges of this technology in healthcare. In this paper, a fuzzy logic-based secure hierarchical routing scheme using the firefly algorithm (FSRF) is presented for IoT-based healthcare systems. FSRF comprises three main frameworks: fuzzy trust framework, firefly algorithm-based clustering framework, and inter-cluster routing framework. A fuzzy logic-based trust framework is responsible for evaluating the trust of IoT devices on the network. This framework identifies and prevents routing attacks like black hole, flooding, wormhole, sinkhole, and selective forwarding. Moreover, FSRF supports a clustering framework based on the firefly algorithm. It presents a fitness function that evaluates the chance of IoT devices to be cluster head nodes. The design of this function is based on trust level, residual energy, hop count, communication radius, and centrality. Also, FSRF involves an on-demand routing framework to decide on reliable and energy-efficient paths that can send the data to the destination faster. Finally, FSRF is compared to the energy-efficient multi-level secure routing protocol (EEMSR) and the enhanced balanced energy-efficient network-integrated super heterogeneous (E-BEENISH) routing method based on network lifetime, energy stored in IoT devices, and packet delivery rate (PDR). These results prove that FSRF improves network longevity by 10.34% and 56.35% and the energy stored in the nodes by 10.79% and 28.51% compared to EEMSR and E-BEENISH, respectively. However, FSRF is weaker than EEMSR in terms of security. Furthermore, PDR in this method has dropped slightly (almost 1.4%) compared to that in EEMSR.

The Internet of Things (IoT) is a new platform for creating global communications between billions of devices around the world. IoT and wireless sensor networks (WSNs) are heavily connected to each other because sensors installed on physical objects sense and collect data from the environment, and then process and send it to the base station (BS)[1,2]. Therefore, sensors are important and vital elements in IoT. A WSN-based IoT network is made up of small sensors that measure the environment and collaborate with each other to gather information about the environmental status and send it to the BS or sink node[3,4]. IoT technology can be used in a variety of applications, including healthcare and elderly care. Smart healthcare is an important aspect of human life around the world, and it is expected that the technology will earn several billion dollars in the near future[5,6].

[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]School of Computing, Gachon University, 1342 Seongnamdaero, Seongnam 13120, South Korea. [4]Department of Information Systems, College of Economics and Political Science, Sultan Qaboos University, Al Khoudh, Muscat, Oman. [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]Department of Information Technology, University of Human Development, Sulaymaniyah, Iraq. [8]Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, Taiwan. ✉email: rahmania@yuntech.edu.tw; liliatightiz@gachon.ac.kr

Unfortunately, the continuous aging of the population and chronic diseases have pressurized modern healthcare systems and increased demands for hospital beds, doctors, and nurses[7,8]. For this reason, it is necessary to present a solution to reduce pressure on healthcare systems as well as provide high-quality services to patients. COVID-19 has recently revealed the importance of rapid, comprehensive, and accurate electronic healthcare, including medical and physiological data to detect coronavirus accurately[9,10]. Thus, the use of emerging technologies like IoT in healthcare systems help identify patients and provide conditions for supervising disease treatment and obtaining new evaluations simultaneously. Furthermore, this technology can act as a potential solution to reduce pressure on healthcare systems[11,12]. Figure 1 shows IoT applications in smart healthcare.

The aim of healthcare monitoring is to track the patient's body parameters and provide fixed and reliable data to physicians or medical teams to better diagnose diseases. This approach will be particularly a great help to patients and elderly users when needing medical services in an unexpected and dangerous situation[13,14]. In smart healthcare, IoT sensors are mainly designed at low cost, low energy consumption, ease of setting up, and stable connectivity. They are responsible for collecting and processing vital data such as electrocardiogram (ECG), oxygen blood saturation, blood pressure (BP), heart rate, blood sugar, pulse rate, brain activity, temperature, and humidity[15,16]. WSN-based IoT networks face energy challenge due to the presence of sensor nodes with limited sources, especially energy. Therefore, obtaining the longest network lifetime is of great significance. Clustering is a successful solution to designing energy-efficient routing schemes because it increases scalability and maintains bandwidth[17,18]. In a clustered network, a sensor node will locally communicate with its cluster head node (CH). In this network, communication with BS is done only through CHs. On the other hand, wearable healthcare applications distribute personal and private data. In this case, security threats such as denial of services (DoS) attacks are carried out by Internet hackers[19,20]. Therefore, hostile nodes obtain and analyze medical data. For this reason, security requirements such as privacy and data integrity should be provided against invaders. In this situation, it is important to create a secure connection between IoT nodes[21,22]. Given that IoT includes heterogeneous devices, security must be guaranteed even for simple devices such as sensors. Therefore, it is important to have a trust framework that prevents the choice of high-risk nodes as intermediate nodes in the routing path. Designing secure and energy-efficient routing protocols to secure data transfer to IoT health devices is a challenging task[23,24].

In this paper, a fuzzy secure hierarchical routing scheme based on the firefly algorithm (FSRF) is suggested for WSN-based IoT networks. FSRF seeks to achieve two goals, namely improving network security and increasing energy efficiency. Note that security and energy consumption are inversely related to each other because powerful security frameworks usually consume a lot of energy. To solve this challenge in IoT, a secure energy-efficient routing approach must be designed to consider both energy efficiency and trust levels in different phases. FSRF consists of three main frameworks: fuzzy trust framework, firefly algorithm-based clustering framework, and inter-cluster routing framework. The main contributions of this paper are as follows:

- In FSRF, a fuzzy theory-based trust framework is offered to evaluate the trust of IoT nodes and counteract cybersecurity attacks against IoT networks. This framework must be able to detect and prevent various attacks such as black hole, flooding, wormhole, sink hole, and grey hole. Four scales, namely the packet delivery ratio (PDR), packet transfer frequency (PTF), packet reception frequency (PRF), and the consumed energy ratio (ECR) are considered to design this trust framework.
- In FSRF, a clustering method based on the firefly algorithm is offered to lower the energy consumed by IoT nodes, communication overhead, and congestion on the network. In this clustering technique, a new objective

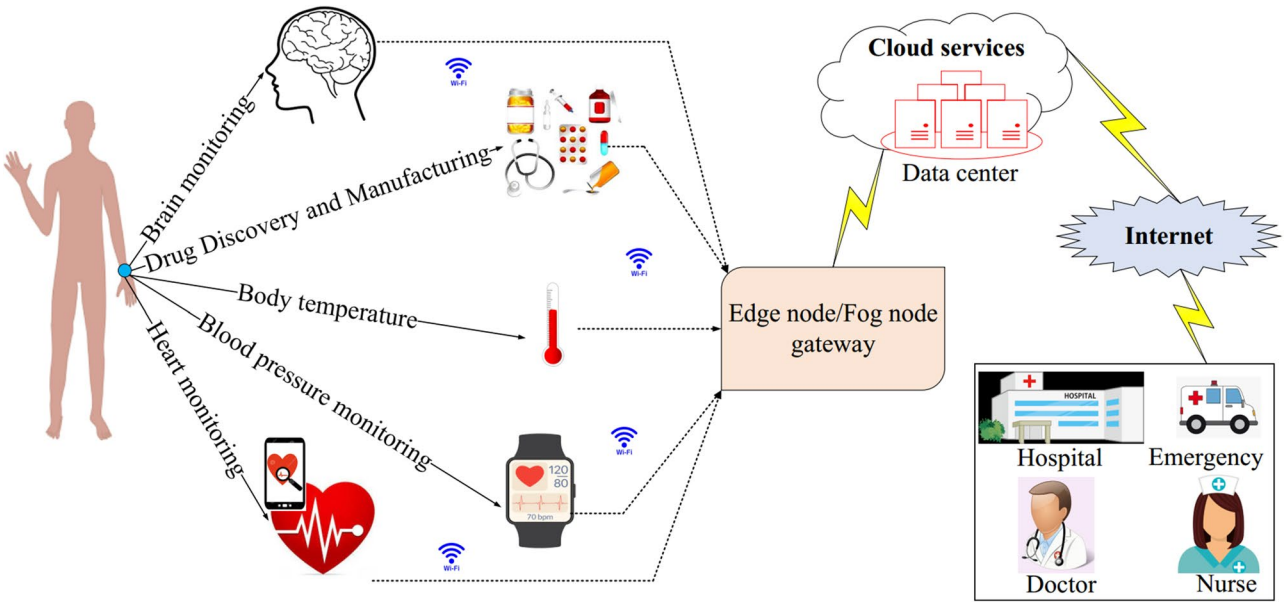

Figure 1. IoT applications in smart healthcare.

function is suggested. This function helps improve network security in the clustering process because it includes the trust level of IoT nodes. Also, paying attention to the energy level and the number of hops between CHs and the base station in this objective function has improved energy efficiency in this method.
- In FSRF, a routing method is offered to find reliable and energy-efficient routes between nodes to reduce the risk of forming fake and unsafe paths in the network because in this scheme, reliable nodes participate in the route discovery process and hostile nodes are not allowed to cooperate in this process.
- In this paper, FSRF is compared with EEMSR and E-BEENISH with regard to network longevity, energy consumption, and packet delivery ratio. This comparison shows that FSRF well guarantees energy efficiency in the network because it increases the network longevity by 10.34% and 56.35% and the energy level stored in the nodes by 10.79% and 28.51% compared to EEMSR and E-BEENISH, respectively. However, FSRF is weaker in terms of security than EEMSR and it has less PDR (almost 1.4%) than EEMSR.

The structure of the paper is as follows: "Related works" section examines some research on cyber security attacks on IoT and their countermeasures. In "Base concepts" section, the main concepts used in the FSRF, namely the firefly algorithm (FA) and fuzzy logic are expressed. "System model" section includes the network model, the energy model, and the attack model. "The proposed method" section has stated various steps of FSRF. Simulation and evaluation results are presented in "Simulation and result evaluation" section. The conclusions obtained from this paper are described in "Conclusion" section.

Related works

In[25], a security framework for routing is provided to prevent cyber security attacks in the Industrial Internet of things (IIoT). For designing this framework, the authors have benefited from different technologies like software-defined networking (SDN), network function virtualization (NFV), and blockchain. The designed framework is flexible, programmable, and secure. Moreover, a three-level SDN/NFV framework is employed in each domain to control and plan desired forwarding devices when calculating optimal routing policies. On the other hand, SDN controllers employ a blockchain framework to build a reliable environment. Next, a secure routing scheme based on this framework is introduced to support node authentication and behavior authentication in the network. The simulation process is done on the OMNET++ platform. The results show that the proposed system is better than other schemes with regard to scalability and stability when occurring attacks.

In[26], a reliable and efficient route selection solution called REERS is offered to get better energy efficiency and lower delay in IoT applications. Initially, REERS employs a clustered data aggregation model that regards energy levels for selecting cluster heads among IoT devices. Next, CHs collect the sensed data and delete duplicated information. Thereafter, various routes are found to the destination for transferring aggregated data packets. Finally, these packets will be directed with the lowest consumed energy, less hop count, and less lost data. The experimental results show that REERS optimizes delay, network longevity, as well as throughput, and PDR.

In[27], the authors have introduced a blockchain-based lightweight authentication structure to check the validity of ordinary sensors. Note that IoT sensors have a short lifetime due to energy restrictions, thus they require a small validity value in blockchain to obtain a lightweight authentication structure. The network controller employs a genetic algorithm-based software to calculate paths. In addition, an on-demand routing technique is applied to optimize the energy used by nodes. The authors have suggested a path-checking framework to investigate the existence of malicious nodes. Furthermore, a novel structure is introduced to limit the activity of the hostile nodes. A list of hostile nodes is stored in the blockchain. This list is employed by the path-checking structure. The experimental results indicate the successful performance of this method with regard to consumed energy and the detection rate of hostile nodes.

In[28], a routing technique based on the shuffled frog-leaping algorithm (SFLA) called RISA is provided for the Internet of Things. This technique employs SFLA to pick out a content-based path from source to destination. Content-based routing decreases the number of transmitted data packets and redundancy through data aggregation. This operation has a great impact on maintaining network resources. When a data packet moves from source to destination, the shortest and most optimized path must be selected to minimize energy consumption. RISA regards energy efficiency and consequently improves network longevity since it applies an appropriate data aggregation technique. Simulation results in MATLAB software show that RISA can optimize energy consumption, network longevity, throughput, and PDR.

In[29], a Harris Hawks Optimization (HHO)-based reliable data dissemination technique called RDDI is suggested for IoT. This secure data dissemination framework offers a fuzzy hierarchical network model for WSN-based IoT networks. RDDI detects attacks and supervises information exchanged between nodes. It builds the data transfer process based on the energy and geographic location of nodes to improve the routing capability. Additionally, it employs a fuzzy clustering structure to pick out a trusted path. The authors evaluated RDDI with regard to five criteria, including reliability, end-to-end delay, consumed energy, computational overhead, and packet-sending distance in different multi-cluster scenarios. The experimental results emphasize the successful performance of RDDI in comparison with other approaches with regard to energy consumption, reliability, end-to-end delay, and computational overhead.

In[18], a tree-based secure routing scheme supported by a dragonfly algorithm called CTSRD for IoT-based smart agriculture. It employs a decentralized and light trust structure called W-Trust. This structure regards a punishment coefficient to decrease the trust of hostile nodes. In contrast, it grows the trust of the valid nodes according to a growth coefficient. Furthermore, it makes a trusted clustering framework (T-Clustering). In this framework, CHs are selected from valid nodes. Eventually, CTSRD builds an inter-cluster routing tree inspired by the dragonfly algorithm named DA-Tree, which is safe, sustainable, and optimal and balances the energy used in the network and extend the longevity of the network. The experimental results emphasize that CTSRD is able

to distribute consumed energy uniformly in comparison with other approaches and consequently gets better network longevity. However, PDR in this scheme is low.

In[30], a multi-level trusted energy-efficient routing scheme called EEMSR in IoT. The authors have used a clustering technique in this method since EEMSR is a effective solution in terms of energy consumption and scalability. In EEMSR, the analytic hierarchy process (AHP) is employed to forecast accurate weight coefficients in the normalization operation. Furthermore, an enhanced genetic algorithm (GA) is suggested to get the best performance and decide on intermediate nodes in multi-hop paths. This enhanced GA lowers the consumed energy in the network and counteracts the weaknesses of GA in the routing process. In addition, a multi-trust framework has been employed to defend against different attacks on the network. This framework computes the trust coefficient in the clustering and routing process. This coefficient is obtained from three trust values, including data perception, data fusion, and communication trust. EEMSR has also focused on both energy efficiency and security. The experimental results emphasize the successful performance of EEMSR compared to other schemes.

In[31], an energy-efficient routing technique named E-BEENISH is presented for heterogeneous WSNs. It analyzes the energy used in inter-cluster and intra-cluster communications to balance energy consumption. E-BEENISH regards a weighted probability for each node when choosing CHs. This probability relies on remaining energy and the distance from the sink node to the desired node. E-BEENISH introduces a simple algorithm that considers the distance between the desired node and BS to overcome the threshold settings in BEENISH. The authors also studied the effect of the heterogeneity of sensor nodes in terms of energy used in the network. Simulation results show that E-BEENISH gets better network longevity in comparison to other clustering protocols.

In WSN-based IoT networks, sensor nodes include various constraints, especially energy, memory, and computing power. These constraints have faced challenges such as shortening the network lifetime and increasing the lost data packets. Therefore, the necessity of a hierarchical routing method in these networks is increasingly evident because clustering is a successful solution for designing energy-efficient routing methods. According to research works studied in this section and Table 1, it can be seen that in recent years, many hierarchical routing methods, for example, REERS[26] and E-BEENISH[31] have been provided in wireless sensor networks to manage energy consumption in these networks. However, these methods do not pay attention to the security issue and, if there are hostile nodes on the network, they will face weak performance. Note that sensor nodes can be deployed in a hostile environment and may be easily captured by attackers, which prevent the proper network performance by doing hostile operations. In this condition, it is important to create a secure connection between IoT nodes. Hence, many researchers focus on powerful security methods, such as Cao et al.[25] and Abbas et al.[27]. These methods are often not suitable for WSNs because they do not pay attention to the energy constraints of the nodes, which can reduce network longevity and lose their normal performance. This shows that the design of secure and energy-efficient routing protocols is a very important issue. Among the methods studied in this paper, some researchers, for example, RDDI[29], CTSRD[18], and EEMSR[30] have taken into account both energy efficiency and security. However, research on hierarchical and secure routing methods is still known as an important research

Method	Publication year	Security mechanism	Routing technique	Energy efficiency	Strengths	Weakness
Cao et al.[25]	2021	Blockchain	An SDN-based secure routing	×	Designing a flexible, programmable, and secure routing framework	Not paying attention to the energy index in the routing process
REERS[26]	2021	×	Clustering routing method	✓	Increasing energy efficiency and reducing delay	Not having a security mechanism
Abbas et al.[27]	2021	Blockchain	A GA-based routing protocol	×	Designing a lightweight authentication structure, ability to accurately detect attacking nodes	Not paying attention to the energy index in the routing process, high execution and transaction costs
RISA[28]	2020	×	A SFLA-based content centric routing method	✓	Increasing energy efficiency	Not designing a security mechanism
RDDI[29]	2020	HHO-based watchful node selection process	A hierarchical energy-aware geographic routing based on the fuzzy clustering	✓	Enhancing energy efficiency, detecting and isolating malicious nodes	Not evaluating its robustness and efficiency against cyber-security attacks
CTSRD[18]	2023	A decentralized and light trust structure called W-Trust	A tree-cluster based routing scheme supported by a dragonfly algorithm	✓	Considering energy efficiency, using a tree-cluster network topology	Low packet delivery rate (PDR)
EEMSR[30]	2021	A multi-trust framework based on data perception trust, data fusion trust, and communication trust	AHP-based clustering and a GA-based routing protocol	✓	Balancing energy consumption in the network, designing a strong security mechanism, detecting malicious nodes	High time complexity
E-BEENISH[31]	2019	×	A clustering routing technique	✓	Considering remaining energy and the distance from the sink node for selecting CHs, high scalability	Not designing a security mechanism
FSRF	×	A fuzzy trust mechanism	A clustering routing method based on firefly algorithm	✓	High network lifetime, high scalability, considering energy efficiency, using a strong security mechanism	Low packet delivery rate (PDR)

Table 1. Comparison of the related works.

gap, which requires further investigation and analysis. In this paper, a fuzzy secure hierarchical routing scheme based on the firefly algorithm (FSRF) is proposed for WSN-based IoT networks. In FSRF, a fuzzy theory-based trust structure is provided to evaluate the trust of IoT nodes and counteract cybersecurity attacks against IoT networks. Furthermore, this method includes a clustering method based on the Firefly algorithm that improves network security in the clustering process because it considers the trust levels of the nodes, and improves energy consumption in the network because it pays attention to the energy level and the number of hops to the base station. FSRF presents a routing method for finding reliable and energy-efficient routes between nodes in the network.

Base concepts

FSRF employs a nature-inspired optimization algorithm named the firefly algorithm (FA) and fuzzy logic (FL). Therefore, these two techniques are explained in this section.

Nature-inspired optimization algorithms. These algorithms are rooted in the social behaviors of biological species, for example, birds, fish, ants, and fireflies. In this system, the behaviors of agents that interact locally with their environment have led to the emergence of coherent global patterns. These algorithms have benefited from self-organization, parallel operations, distributed operations, flexibility, and stability[32,33]. For this reason, their applications have gradually expanded to solve many IoT issues, including routing such as[18,28,29], and[34]. They are used to solve many real-world engineering issues. For example, the firefly algorithm (FA) is a nature-inspired optimization algorithm presented by Yang in 2007[35]. This algorithm has benefited from flexibility in the population. This means that it is scalable. It can perform a relatively large search and corrects the responses in the search process. When getting the optimal solution, FA makes a balance between exploration and exploitation, and eventually reaches an optimal global behavior. These advantages have led to the use of this technique in the proposed method. This algorithm is inspired by the brightness of the fireflies. In the FA algorithm, there are two important issues: light intensity changes and formulation of brightness and attractiveness. It is assumed that the attractiveness of fireflies is proportional to their brightness. Moreover, the brightness is determined by an objective function. In this algorithm, the less-light firefly will be attracted to the high-light firefly[35].

Fuzzy logic. Fuzzy logic is a good scheme for mapping from the input space into an output space. It is a precise method based on approximate and inaccurate data. Fuzzy systems (FSs) are defined by fuzzy set theory[36,37]. A fuzzy set is a borderless set, which includes elements with partly membership degree (usually between 0 and 1). The fuzzy system may be less accurate than conventional systems, but more like our everyday experiences as human decisions. The fuzzy inference is the mapping process from a given input to an output using FL. Then, this mapping provides a basis for decision-making. The fuzzy inference comprises membership functions (MFs), fuzzy operators, and IF-THEN rules. Mamdani and Sugeno are two common fuzzy inference systems. A Mamdani system states that the output MFs are fuzzy sets. After aggregating the results, there is a fuzzy set for each output variable, which requires a defuzzifier. A Sugeno system also supports this behavior and is very similar to the Mamdani system. In fact, the first two parts of the fuzzy inference process, the fuzzifier process, and the use of fuzzy operators are similar. The main difference between Mamdani and Sugeno models is that output MFs in Sugeno can be fixed or linear. Whereas, output MFs in Mamdani are nonlinear[36,37].

Usually, FS has four components: fuzzifier, fuzzy rules, inference engine, and defuzzifier. Fuzzification must categorize numerical scales into fuzzy sets. The knowledge base consists of IF-THEN rules that show linguistic reasoning. An inference engine executes the rules on the fuzzy inputs to get fuzzy results. The fuzzy controller needs the knowledge of an expert or operator experience to determine appropriate control rules and MFs. Fuzzifier can convert crisp data or fuzzy data into appropriate linguistic values through language variables and a variety of MFs, such as triangular, trapezoidal, and Gaussian. MF maps from each element of the input variables to a membership degree between 0 and 1. Triangular functions are usually used in FSs because of their simplicity. Finally, defuzzifier determines how to extract the crisp value from the fuzzy set. The well-known defuzzifier is the centroid, which shows more reliable results than others. The selection of a defuzzifier is very important and has a significant impact on the speed and accuracy of the fuzzy model[38].

System model

This section describes different parts of the system model namely the network model, the energy model, and the attack model.

Network model. In FSRF, the network is made up of heterogeneous IoT nodes ($n_1, n_2, ..., n_i, ..., n_T$, so that T indicates the total number of nodes in the network) and a base station. The BS has different responsibilities, like data analysis and decision-making about data received from cluster heads. It is a motionless node, and all nodes know its position on the network. A special identifier is employed by each node (i.e. n_i). The distribution of the nodes in the network is done using a random manner. In FSRF, there is a connection between network nodes and the global positioning system (GPS). Hence, the nodes can obtain their spatial coordinates in the network. FSRF regards heterogeneous nodes, which have different energy levels, computational power, and storage capacity. In FSRF, the nodes are placed in clusters. Each cluster is made up of a cluster head (CH) and a number of cluster members (CMs). CMs have various responsibilities such as sensing the environment and transferring the data to the CH. They perform the intra-cluster data transfer operation using a single-hop manner. In addition, CHs have one important responsibility i.e. gathering data from its CMs and transferring the aggregated data to BS.

They perform the inter-cluster data transfer operation using a multi-hop manner to transmit their data to the base station. See the network model in Fig. 2.

Energy model. The data transfer operation, which means sending and receiving data, is known as the most serious factor of energy consumption in the network. FSRF regards both free space and multi-path models to control how much energy is consumed in the recipient and sender. Note that the energy model used in this paper is similar to the energy radio model proposed by Heinzelman et al.[39]. In this case, the whole energy used in all nodes is equal to their consumption energy when sending and receiving data.

$$E_{Total} = \sum_{i=1}^{N} \left(E_{tx}^{i} + E_{rx}^{i}\right) \tag{1}$$

Here, E_{tx}^{i} and E_{rx}^{i} express the required energy of n_i for transferring and getting data, respectively.

When two nodes want to exchange their data with each other. In this case, one of them acts as a transmitter (also called n_t) and the other node plays the role of a receiver (also called n_r). Suppose l indicates the size of the exchanged data and $d = \sqrt{(x_r - x_t)^2 + \left(y_r - y_t\right)^2}$ is the distance from n_t to n_r where $\left(x_r, y_r\right)$ and $\left(x_t, y_t\right)$ are the spatial coordinates of n_r to n_t, respectively. In this case, Eq. (2) determines how much energy is used by n_t.

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

So that E_{elec} is the energy needed for the electrical equipment of n_t or n_r, ε_{fs} represents the amplification factor in the free space, and ε_{mp} indicates the amplification factor in the multi-path space. In Eq. (2), if the distance between n_t and n_r (i.e. d) is shorter than d_0 (i.e. the boundary value), the consumed energy is calculated based on the free-space model (the first line of Eq. (2)); otherwise, it is calculated based on the multipath model (the second line of Eq. (2))[39]. d_0 is obtained from Eq. (3), which demonstrates a boundary condition for the data transfer scheme employed by n_t and n_r.

$$d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}} \tag{3}$$

Finally, Eq. (4) determines how much energy is used by n_r:

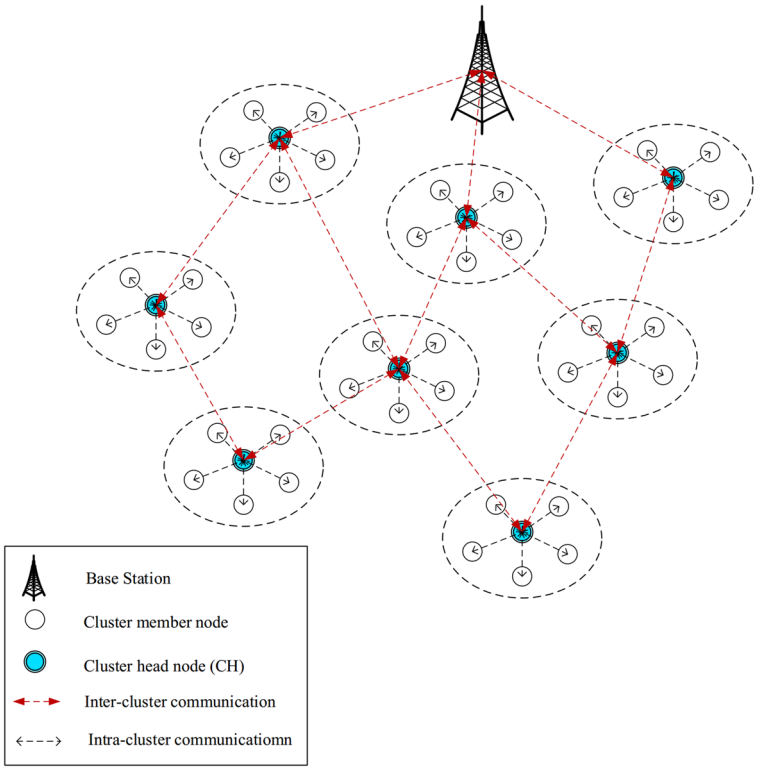

Figure 2. Network model in FSRF.

$$E_{rx}(l) = l \times E_{elec} \tag{4}$$

Attack model. IoT employs wireless channels to communicate between the network nodes. Therefore, this network is exposed to serious security harm. The invading nodes can penetrate the network in different ways and launch various attacks on the network. This ruins the normal network performance and affects the secure data transfer operation due to the removal or manipulation of data packets and the energy discharge of the IoT nodes. Therefore, it is necessary that the nodes involved in the process are safe and reliable. FSRF focuses on blackhole, sinkhole, wormhole, selective forwarding, and flooding.

- *Black hole or sinkhole attacks:* In these attacks, when an invading node (black hole or sinkhole) gets a route request from other network nodes, it replies to this message to state that it has a suitable path to the destination, even though this claim is not right. When the requested node obtains this response, it may employ the insecure path, which includes a black hole or sinkhole, for transferring data. In this case, the invading node eliminates all data packets received from the source node.
- *Wormhole attack:* In this attack, two invading nodes build a tunnel and state that they are neighbors (i.e. they are very close together) while this claim may be wrong. High-power nodes may carry out this attack. In this case, they have more resources than normal nodes. When the invading nodes build a forged path, they seek to attract network traffic and declare that the path is very efficient and suitable, and has smaller hops to the BS while it is not fact. After attracting the data traffic of normal nodes, the invading nodes can track their communications, copy and manipulate their data packets.
- *Selective forwarding attack:* This attack, also called grey hole, is an advanced model of black hole attacks. In this attack, the invading node selectively deletes some packets but not all of them. It deletes only packets transmitted to a specific destination or eliminates a special type of packets.
- *Flooding attack:* In this attack, the invading node continually transmits route requests to a specific node. This work leads to the discharge of the target node, its storage space is full. This is because the invading node misuses the fact that some information about the route requests is stored in the memory of the target node. In this case, the target node cannot respond to the legal requests of other node and dies quickly because it loses high energy.

The proposed method

In this section, a fuzzy secure hierarchical routing scheme based on the firefly algorithm (FSRF) is explained for WSN-based IoT networks. FSRF comprises three main frameworks: fuzzy trust framework, firefly algorithm-based clustering framework, and inter-cluster routing framework.

Fuzzy trust framework. In FSRF, the fuzzy trust framework is tasked to analyze the reputation of nodes according to their interactive behavior when transferring and receiving data packets. Determining the trust value of the network nodes will be done using a fuzzy trust framework. Algorithm 1 offers a pseudo-code of the fuzzy trust framework. The Mamdani fuzzy system is used to design this framework. It comprises two inputs (i.e. direct trust and indirect trust), an output (i.e. total trust of network nodes), and the rule base. Each IoT node executes this fuzzy framework to characterize the trust value of its neighboring nodes. Additionally, the trust value of the nodes changes dynamically, and their energy decreases, and some of them die. Therefore, IoT nodes must renew the trust values related to their neighboring nodes at certain time intervals.

Fuzzy inputs. The proposed fuzzy framework comprises two inputs called direct trust and indirect trust.

- *Direct trust of n_i related to n_j (T_{ij}^{direct}):* T_{ij}^{direct} indicates the urgent trust generated by n_i for n_j. It is acquired through the direct connection of n_i and n_j. In FSRF, the direct trust regards the packet delivery ratio (PDR), packet transfer frequency (PTF), packet reception frequency (PRF), and the consumed energy ratio (ECR).

  - PDR_j means a packet reception rate corresponding to n_j. It expresses the ratio of packets received by n_j to all data packets sent to this node. Note that a high PDR confirms the successful performance of n_j and shows its reliability. However, if n_j does not experiences a suitable PDR, it means that n_j has high missing data. In this case, n_j may be an invader. This increases the probability of attacks such as black holes, sink hole, and grey hole. Hence, the trust value corresponding to n_j decreases. PDR_j is calculated through Eq. (5).

$$PDR_j = \frac{M_j^{received}}{M_j^{total}} \tag{5}$$

    Here, $M_j^{received}(t)$ and M_j^{total} represent the number of packets received and sent to n_j, respectively.
  - PTF_j determines how many packets are transferred by n_j at the time interval $[t, t + \Delta t]$. Note that a high PTF means that n_j may be an invader because there is a high likelihood of flooding or wormhole attacks. In this case, the trust value of n_j is reduced. PTF_j is calculated by Eq. (6).

$$PTF_j = \frac{M_j^{Transferred}}{\Delta t} \tag{6}$$

where $M_j^{Transferred}$ counts how many packets are transferred at the interval $[t, t + \Delta t]$.

- PRF_j determines how many packets are received by n_j at the time period $[t, t + \Delta t]$. Note that a high PRF_j confirms that n_j is safe and reliable. However, if n_j gets low PRF_j, n_j may be an invader because the probability of attacks such as black hole, sinkhole or gray hole is high. PRF_j can be achieved through Eq. (7).

$$PRF_j = \frac{M_j^{Received}}{\Delta t} \tag{7}$$

So that $M_j^{Received}$ counts the packets received in the interval $[t, t + \Delta t]$.

- ECR_j determines how much energy is consumed by n_j in the time period $[t, t + \Delta t]$. Note that a high ECR_j states that n_j may be an invader because the probability of a flooding attack is high. ECR_j is determined through Eq. (8).

$$ECR_j = \frac{E_j(t) - E_j(t + \Delta t)}{\Delta t} \tag{8}$$

So, $E_j(t)$ and $E_j(t + \Delta t)$ are the residual energy of n_j in two times t and $t + \Delta t$, respectively.

According to the stated parameters, T_{ij}^{direct} is calculated based on Eq. (9).

$$T_{ij}^{direct} = \frac{\lambda_1 PDR_j + \lambda_2 PRF_j}{\lambda_3 PTF_j + \lambda_4 ECR_j} \tag{9}$$

So, λ_1, λ_2, λ_3, and λ_4 are the weight coefficients adjusted in $[0, 1]$ and $\sum_{i=1}^{4} \lambda_i = 1$. In FSRF, the window mean with exponentially weighted moving average (WMEWMA) is employed to renew T_{ij}^{direct}. It applies the window length w to consider the historical trust amounts when calculating T_{ij}^{direct}. Hence, n_i does not only rely on the present amount and uses the set of trust values to better decide on T_{ij}^{direct}. As a result, Eq. (10) renews T_{ij}^{direct}.

$$T_{ij}^{direct}(l) = (1 - \beta)\frac{\sum_{k=l-w}^{l-1} T_{ij}^{direct}(k)}{w} + \beta T_{ij}^{direct}(t) \tag{10}$$

So that β is a coefficient adjusted in $[0, 1]$. Membership function (MF) related to T_{ij}^{direct} is depicted in Fig. 3. T_{ij}^{direct} contains three modes, low, medium, and high.

- *Indirect trust of n_i related to n_j* ($T_{ij}^{indirect}$)**:** $T_{ij}^{indirect}$ characterizes the trust amount obtained from the recommended nodes, which are the common and reliable neighbors between n_i and n_j. The recommended nodes should be picked out from reliable nodes whose trust amount is more than $T_{threshold}$. We assume that there is a set called R, which includes p recommended nodes between n_i and n_j so that $R = \left\{ n^1_{Recommender}, n^2_{Recommender}, ..., n^p_{Recommender} \right\}$. In this case, $T_{ij}^{indirect}$ is obtained using Eq. (11).

$$T_{ij}^{indirect} = \frac{1}{p} \sum_{x \in R}^{p} \left(T_{ix}^{direct} \cdot T_{xj}^{direct} \right) \tag{11}$$

where T_{ix}^{direct} and T_{xj}^{direct} express the direct trust of n_i to n_x and the direct trust of n_x to n_j, respectively. Also, n_x indicates a recommended node. The MF of $T_{ij}^{indirect}$ is displayed in Fig. 4. $T_{ij}^{indirect}$ contains three modes, including low, medium, and high.

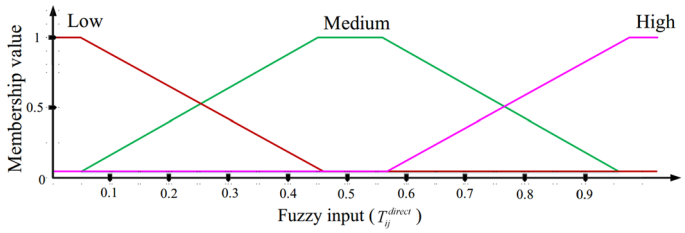

Figure 3. MF related to T_{ij}^{direct}.

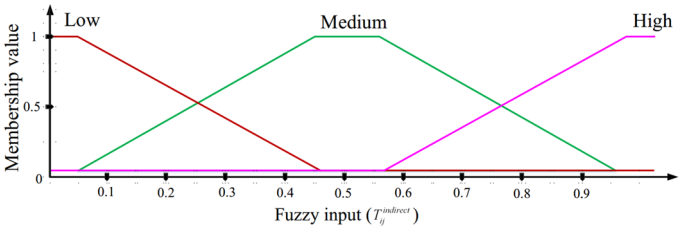

Figure 4. MF of $T_{ij}^{indirect}$.

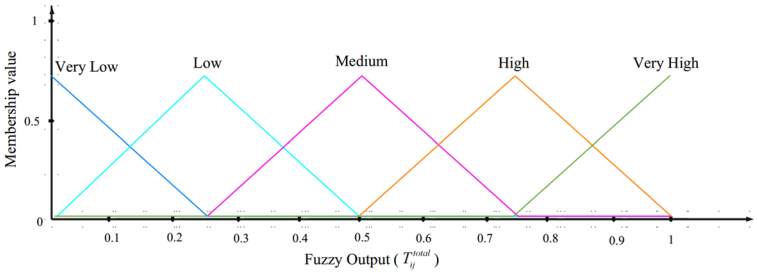

Figure 5. MF of T_{ij}^{total}.

Number	T_{ij}^{direct}	$T_{ij}^{indirect}$	T_{ij}^{total}
1	Low	Low	Very low
2	Low	Medium	Low
3	Low	High	Medium
4	Medium	Low	Low
5	Medium	Medium	Medium
6	Medium	High	High
7	High	Low	Medium
8	High	Medium	High
9	High	High	Very high

Table 2. Rule base.

Fuzzy output. The output of this fuzzy trust framework illustrates the total trust (T_{ij}^{total}), which includes five modes (very low, low, medium, high, and, very high). See the MF of T_{ij}^{total} in Fig. 5.

Rule base. The proposed trust framework defines the rules presented in Table 2. For example, Rule 1 is stated below.

Rule 1: **IF** T_{ij}^{direct} is *low* **AND** $T_{ij}^{indirect}$ *low* **THEN** T_{ij}^{total} is *very low*.

Algorithm 1 Fuzzy logic-based trust model

Input: n_i, n_j: two network nodes
Output: T_{ij}^{total}: Total trust of n_j calculated by n_i
Begin
1: n_i: Compute the packet receive rate of n_j (PDR_j) based on Equation 5;
2: n_i: Compute the packet transmission frequency of n_j (PTF_j) according to Equation 6;
3: n_i: Determine the packet receive frequency of n_j (PRF_j) using Equation 7;
4: n_i: Obtain the energy consumption rate of n_j (ECR_j) from Equation 8;
5: n_i: Compute the direct trust of n_j (T_{ij}^{direct}) using Equation 9;
6: n_i: Refresh T_{ij}^{direct} based on the WMEWMA method presented in Equation 10;
7: n_i: Generate the fuzzy value of T_{ij}^{direct} by using the fuzzy membership function presented in Figure 3;
8: n_i: Compute the indirect trust value of n_j ($T_{ij}^{indirect}$) by recommender nodes based on Equation 11;
9: n_i: Produce the fuzzy value of $T_{ij}^{indirect}$ based on the fuzzy membership function displayed in Figure 4;
10: n_i: Obtain the fuzzy value of the total trust between n_i and n_j (i.e. T_{ij}^{total}) from the proposed fuzzy system;
11: n_i: Calculate the crisp value of T_{ij}^{total} using the fuzzy membership function displayed in Figure 5;
End

Firefly algorithm-based clustering framework. Here, the firefly algorithm-based clustering framework is stated in FSRF. This framework will be executed by the base station. Algorithm 2 describes the clustering framework in FSRF. This framework comprises two steps:

- Clustering
- Cluster maintenance

Clustering. Each IoT node, like n_i, transfers a guide message to BS. The message is named a beacon, which contains the trust amount, location, remaining energy, hops to BS, centrality degree, and communication radius. Next, BS assesses the trust of n_i to differentiate the trusted nodes from untrusted nodes. Note that the trusted nodes have high trust (more than $T_{threshold}$). In FSRF, only trusted nodes can be cluster heads. Then, the BS starts the FA-based CH selection algorithm. In this algorithm, each firefly plays the role of an IoT node (n_i), and the value of this firefly states the chance of n_i to behave as CH between neighboring nodes. In the first step, the value of each firefly is a random number. Here, each n_i expresses a firefly, which may be a CH. It presents a response to the CH selection problem. The primary attractiveness of the fireflies is displayed as β_0 determined by the RAND function. In this CH selection framework, BS has an important responsibility, which must employ the FA algorithm to decide on the best CHs in the network. Thereafter, the base station will assess the fitness amount of each response (firefly) based on an objective function. This function is formulated in accordance with five parameters, namely the trust amount, remaining energy, hops to BS, communication radius, and the average distance to the neighboring nodes.

- *Trust amount* (T^{total}): The purpose of this parameter in the objective function is to select secure nodes as a CH because cluster heads have important tasks including the submission of data packets received from CMs, participation in the routing process between CHs, and the transmission of data packets of other CHs to the base station. Therefore, if an insecure node is selected as a CH, it can damage the normal performance of the network. The BS extracts T^{total} from the guide message received from n_i. "Fuzzy trust framework" section explains how to calculate T^{total} in detail. If n_i utilizes a higher trust amount, it has a greater chance to act as a CH because it is safer and can send the data to the network reliably. T^{total} will be normalized by Eq. (12).

$$T_{norm}^{total} = \frac{T^{total} - T_{thershold}}{T_{\max} - T_{thershold}} \tag{12}$$

  So that $T_{thershold}$ and $T_{\max}$ are the minimum acceptable trust determined for CHs and the maximum trust amount in the network, respectively.
- *Remaining energy* (E_r)**:** The purpose of this parameter in the objective function is that energy consumption in the network is evenly distributed between the IoT nodes to increase network longevity. In this regard, high-energy nodes have a responsibility to play the role of cluster heads because CHs have more responsibilities than normal nodes and consume more energy. If low-energy nodes play the role of CH, their energy will be ended quickly. In this case, finding the new CH is also accompanied by a lot of cost, time, and communication overhead. The BS extracts E_r from the guide message obtained from n_i. Energy is very important when deciding on CHs because IoT nodes suffer from energy restrictions on the network. Additionally, network nodes have different amounts of energy. Consequently, if n_i has more energy than other nodes on the network, n_i gets a higher chance to act as CH. E_r is normalized according to Eq. (13).

$$E_r^{norm} = \frac{E_r - E_{\min}}{E_{\max} - E_{\min}} \tag{13}$$

As, E_r, $E_{\min}$, and $E_{\max}$ are the remaining energy of n_i, the lowest energy level, and the maximum energy level of the network nodes, respectively.

- *Hops to BS* (H_c): H_c is very important in the decision-making process for CHs because if CHs have fewer hops to the BS, the data packets reach the BS faster and experience less delay. H_c is normalized using Eq. (14).

$$H_c^{norm} = \frac{H_c - H_{\min}}{H_{\max} - H_{\min}} \tag{14}$$

So that H_c indicates the hop count from n_i to the BS, $H_{\min}$ represents the minimum hops to the BS, where $H_{\min} = 1$, and $H_{\max}$ expresses the maximum hops to the BS, which is dependent on the number of nodes (i.e. N) so that $H_{\max} = N - 1$.

- *Communication radius* (R_{com}): IoT nodes are heterogeneous and R_{com} is different. In FSRF, this subject is intended in the CH selection framework, so that n_i with a large R_{com} gets a greater chance to act as CH because n_i covers a wider range. R_{com} is normalized based on Eq. (15).

$$R_{com}^{norm} = \frac{R_{com} - R_{\min}}{R_{\max} - R_{\min}} \tag{15}$$

So that R_{com}, $R_{\min}$, and $R_{\max}$ are the communication radius of n_i, the least radius, and the highest radius, respectively.

- *Average distance to neighboring nodes* (D_{avg}): D_{avg} shows the centrality of n_i in the cluster. When n_i is close to the cluster center, D_{avg} is low. In this case, if a node close to the cluster center is selected as a CH, this increases energy efficiency in the network because the average distance between this CH and its CMs is reduced and the CH needs less energy to receive the data packets from CMs. This parameter will be calculated using Eq. (16).

$$D_{avg} = \frac{1}{T_i} \sum_{u \in Nei}^{T_i} d(n_i, u) \tag{16}$$

Where T_i indicates the number of neighbors of n_i. Furthermore, $d(n_i, u)$ is the distance from n_i to its neighbor (u). $d(n_i, u)$ is obtained through Eq. (17).

$$d(n_i, u) = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2} \tag{17}$$

Where, (x_i, y_i) and (x_u, y_u) are the positions of n_i and u, respectively. Finally, D_{avg} is normalized using Eq. (18).

$$D_{avg}^{norm} = \frac{D_{avg} - D_{\min}}{D_{\max} - D_{\min}} \tag{18}$$

Where, $D_{\min}$ and $D_{\max}$ are the shortest and longest distances of neighboring nodes on the network, respectively.

As a result, the objective function is calculated in accordance with Eq. (19).

$$F = \frac{\lambda_1 T_{norm}^{total} + \lambda_2 E_r^{norm} + \lambda_3 R_{com}^{norm}}{\alpha_1 H_c^{norm} + \alpha_2 D_{avg}^{norm}} \tag{19}$$

So that λ_1, λ_2, λ_3, α_1, and α_2 represent the weight coefficients where, $\sum_{i=1}^{3} \lambda_i = 1$ and $\sum_{i=1}^{2} \alpha_i = 1$. In Eq. (19), T_{norm}^{total}, E_r^{norm}, R_{com}^{norm}, H_c^{norm}, and D_{avg}^{norm} are normalized in $[0, 1]$ to have the same effect on the objective function.

After determining the chance of each firefly (IoT node), the position of these fireflies is renewed using the firefly algorithm. In the FA-based CH selection framework, the algorithm is repeated 300 times (stop condition). Upon the FA-based clustering framework has ended, the BS picks out the best firefly with the highest fitness amount as a CH. Then, BS informs IoT nodes of their roles. Next, CHs prepare a notification message to inform their neighboring nodes of their roles on the network. This notification message includes the coordinates of CHs. When neighboring nodes get these notification messages, CHs must recognize their members. In the first mode, when an ordinary node gets one or more notification messages from different CHs, it sends a membership request to the nearest CH. In the second mode, when an ordinary node with a trust amount more than $T_{threshold}$ does not get any notification message from CHs, it acts as a CH and broadcasts a notification message to recognize. Now, there are two types of nodes in the network: CHs and CMs. The cluster member nodes are only associated with their CH and transfer their data to it. However, the CHs communicate directly with CMs and their neighboring CH.

Cluster maintenance. The purpose of the cluster maintenance process is to adjust the role of each node through the periodic exchange of guide messages so that the connections can be stable on the network. All IoT nodes participate in the cluster maintenance process. This process includes the following steps:

- *Connect to the cluster:* When a new IoT node is connected to the network, it broadcasts a membership request. A CH node that receives this request faster than other CHs, responds to it and accepts the membership of this new node in its cluster.

- *Leave the cluster:* Each CM examines its connection with its CH through the periodic exchange of guide messages. If this link is invalid, CM has been removed from membership in the cluster, and again, the CM broadcasts a membership request on the network to connect to the nearest CH.
- *Cluster membership checking:* Each CH examines its communication with its CMs through the periodic exchange of guide messages with its CMs. If the communication links are invalid, the CH should cancel the membership of the desired CM in its cluster.
- *Re-clustering:* The status of IoT nodes changes depending on their energy levels or their trust over time. As a result, it is necessary for the BS to constantly monitor their status by periodic guide messages received from IoT nodes. If CHs lose their energy or trust amount, it transfers a cluster update message to its CMs and re-runs the FA-based selection framework in "Firefly algorithm-based clustering framework" section. In this case, the IoT nodes are waiting for the FA-based clustering operation to be implemented by the base station to choose a new CH.

Algorithm 2 FA-based clustering technique

Input: n_i: network nodes, where $i = 1, ..., N$
$N_{fireflies}$: The number of fireflies in the firefly algorithm.
N_{CH}: The number of CH nodes in the network.
T_i: Trust value of n_i
Output: Clustered network
Begin
1: for $i = 1$ to N do
2: n_i: Send a guide message to the base station (BS);
3: if $T_i > T_{threshold}$ then
4: BS: Mark n_i as a trusted node;
5: end if
6: end for
7: BS: Initialize FA factors;
8: BS: Establish the primary firefly population randomly based on the trusted IoT nodes;
9: for $i = 1$ to 300 do
10: BS: Compute the fitness value of each firefly based on T^{total}, E_r, H_c, R_{com}, and D_{avg} using Equation 19;
11: BS: Calculate distance between neighboring fireflies based on the firefly algorithm;
12: BS: Evaluate the attractiveness of neighboring fireflies according to the firefly algorithm;
13: for $j = 1$ to $N_{fireflies}$ do
14: for $k = 1$ to $N_{fireflies}$ do
15: if $F_i < F_j$ then
16: BS: Move firefly i toward firefly j;
17: end if
18: BS: Update new positions of fireflies according to the firefly algorithm;
19: end for
20: end for
21: BS: Sort the fireflies based on the fitness value;
22: BS: Select the firefly with the maximum fitness value as the CH node;
23: end for
24: BS: Inform network nodes of their roles by transmitting a message to them;
25: CH: Broadcast a notification message to the neighboring nodes;
26: n_i: Wait a certain time to get all notification messages from different CHs;
27: if n_i gets several notification messages from different CHs then
28: n_i: Send a membership request to the nearest CH;
29: else
30: n_i: Send a membership request to the desired CH;
31: end if
32: CH: Send an ACK message to n_i for confirming its membership;
33: if n_i does not get any notification message and $T_i > T_{threshold}$ then
34: n_i: Broadcast a notification message to the neighboring nodes;
35: end if
36: CH: Check the membership status of its cluster members (CMs) periodically;
37: if CH does not get any guide message from a CM node then
38: CH: Delete the link between itself and the CM because this link is invalid;
39: CH: Remove this node from its cluster members;
40: end if
41: CM: Check the link status between itself and it CH periodically;
42: if CM does not receive any guide message from its CH then
43: CM: Delete the link between itself and the CH because this link is invalid;
44: CM: Remove its membership in this cluster;
45: CM: Send a new membership request to other CHs in the network;
46: CM: Connect to the nearest CHs in the network;
47: end if
48: for $i = 1$ to N_{CH} do
49: CH: Send a guide message to the base station (BS) periodically;
50: if $T_{CH} < T_{threshold}$ or $E_{CH} < E_{threshold}$ then
51: BS: Send a cluster update message to all cluster member node;
52: go to Line 7;
53: end if
54: end for
End

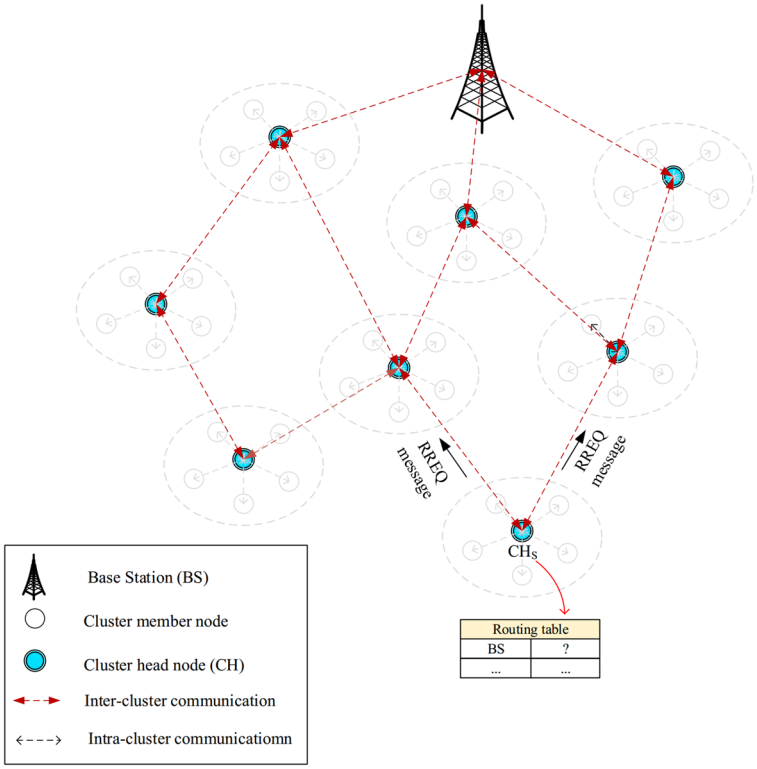

Figure 6. Spreading RREQ message.

MT	H_c	E_R	D_R	T_R
RREQ ID				
Destination IP Address				
Destination Sequence Number				
Source IP Address				
Source Sequence Number				

Figure 7. RREQ message format.

Inter-cluster routing framework. In FSRF, the inter-cluster routing framework consists of two main steps: discovering paths between CHs and maintaining these paths. Algorithm 3 presents the pseudo-code related to the inter-cluster routing framework.

Discovering paths between CHs. FSRF executes an on-demand technique for discovering paths. When a cluster head, like CH_S, is looking for a path to the BS. In the first stage, its routing table is searched, and CH_S examines whether it can find a connected path to the BS. If yes, CH_S connects to the BS through this path to transmit data packets. Otherwise, CH_S begins a path search operation to find a safe and energy-efficient path. In this operation, CH_S creates a route request (RREQ) and spreads it to its one-hop neighbors. View Fig. 6.

See the format of RREQ in Fig. 7. The fields of this message are explained below:

- *Message type (MT):* If $MT = 1$, this control message indicates a RREQ
- $\mathbf{H_c}$**:** It counts hops in the relevant path. CH_S adjusts the initial amount of H_c to zero and then, each intermediate node adds one unit to H_c. In RREQ, H_c helps prevent the formation of routing loops in the created paths.
- *RREQ ID:* Each RREQ is marked with a special ID. This ID and the CH_S address are used for checking RREQs and finding repeated RREQs.
- $\mathbf{E}_R$: It determines how much energy is consumed to send data from CH_S to the desired node through the relevant path. In general, E_R is computed using Eq. (20).

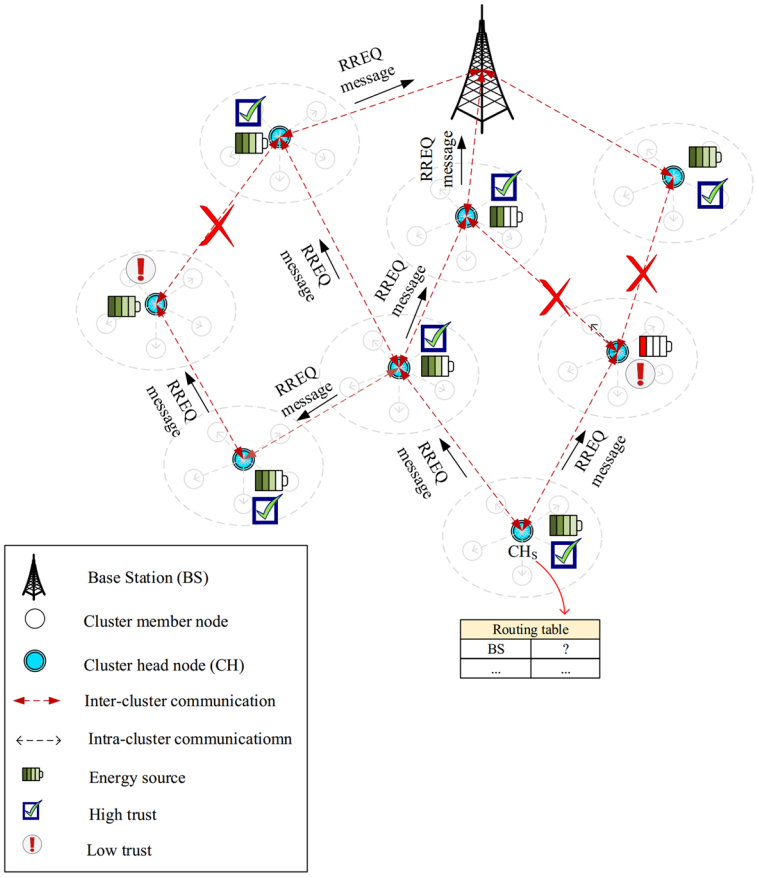

Figure 8. Path search process.

$$E_R = E_{consumed}(Source) + \sum_{CH_{intermediate} \in Route_k}^{Destination-1} E_{consumed}(CH_{intermediate}) + E_{consumed}(Destination) \tag{20}$$

so that $E_{consumed}(Source)$ is the energy used in CH_S for transferring RREQ, $E_{consumed}(CH_{intermediate})$ is the required energy of intermediate nodes to get and forward RREQ, $E_{consumed}(Destination)$ is the required energy of the desired node for getting RREQ. According to the energy model stated in "Energy model" section, Eq. (20) is rewritten as Eq. (21).

$$E_R = E_{tx}^{Source} + \sum_{CH_i \in Route_k}^{Destination-1} \left(E_{tx}^{CH_i} + E_{rx}^{CH_i}\right) + E_{rx}^{Destination} \tag{21}$$

where E_{tx} and E_{rx}, which are respectively obtained from Eqs. (2) and (4), are the required energy for transferring and getting RREQs.

- **D_R:** This field stores the total delay taken from CH_S to the desired node (i.e. base station). The initial amount of D_R is zero. In the next hops, the amount of D_R is dependent on $D_{Propagation}$, $D_{Queuing}$, $D_{Computing}$, and $D_{Transmission}$. In this regard, the value of D_R is refreshed using Eq. (22).

$$D_R = \sum_{i=Source}^{Destination} \left(D_{Propagation}(CH_i) + D_{Queuing}(CH_i) + D_{Computing}(CH_i) + D_{Transmission}(CH_i)\right) \tag{22}$$

So, *Source* and *Destination* are the source and destination nodes, respectively. Note that $D_{Propagation}$ is the time taken for transferring data through the wireless link between two intermediate nodes. There is a direct relationship between $D_{Propagation}$ and the distance between the two nodes. $D_{Propagation}$ is derived from Eq. (23).

$$D_{Propagation} = \frac{Dist\left(CH_i, CH_j\right)}{v_{media}} \tag{23}$$

So that v_{media} is the light speed (i.e. 3×10^8) and $Dist\left(CH_i, CH_j\right)$, which is calculated using Eq. (24), indicates the distance from CH_i to CH_j.

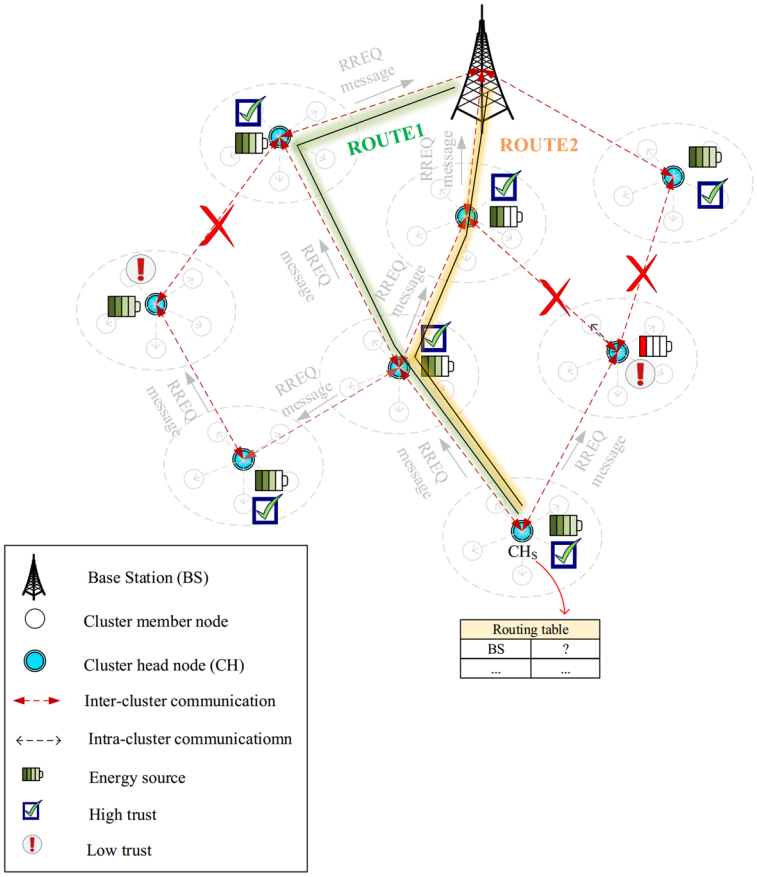

Figure 9. Paths found between CH_S and the BS.

$$Dist\left(CH_i, CH_j\right) = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{24}$$

So that $\left(x_i, y_i\right)$ and $\left(x_u, y_u\right)$ express the positions of CH_i and CH_j, respectively. In addition, $D_{Queuing}$ states a time interval when RREQs have to wait in the buffer queue to send. $D_{Computing}$ represents the time required to process RREQ in the relevant node. In addition, $D_{Transmission}$ indicates the time taken for transferring RREQ to the next CH. It is obtained according to Eq. (25).

$$D_{Transmission} = \frac{msg_{size}}{br} \tag{25}$$

So that br and msg_{size} demonstrate the transfer rate and the size of RREQ, respectively.

- **T_R:** It determines whether a path is reliable. Initially, CH_S adjusts the amount of T_R to its trust amount. Then, T_R will be updated in accordance with Eq. (26) in each hop. The amount of T_R is considered the lowest trust amount of CHs in a routing path. T_R helps CH_S to prevent the selection of insecure paths.

$$T_R = \min_{CH_i, CH_j \in Route_k} \left(T_{ij}^{total}\right) \tag{26}$$

where CH_i and CH_j show the pervious-hop and next-hop intermediate CHs in the present path (i.e. $Route_k$), respectively. Also, T_{ij}^{total} is the trust amount of CH_i to CH_j explained in "Fuzzy trust framework" section.

- *Source IP address (SIA):* It specifies the address of CH_S.
- *Destination IP address (DIA):* It specifies the address of the BS.
- *Source sequence number (SSN):* It helps intermediate nodes to ensure that the information about the reversed path to CH_S is new.
- *Destination sequence number (DSN):* Before choosing a path by CH_S, DSN guarantees that the path is new.

After an intermediate CH gets RREQ, it first examines its ID and ensures that the RREQ is not old. Next, the intermediate node compares its remaining energy (E_r) and its trust amount (T^{total}) with $E_{threshold}$ and $T_{threshold}$, respectively. If $E_r > E_{threshold}$ and $T^{total} > T_{threshold}$, the CH is allowed to rebroadcast the RREQ. Otherwise, this CH should delete the RREQ. This strategy will improve the performance of the routing method in terms of

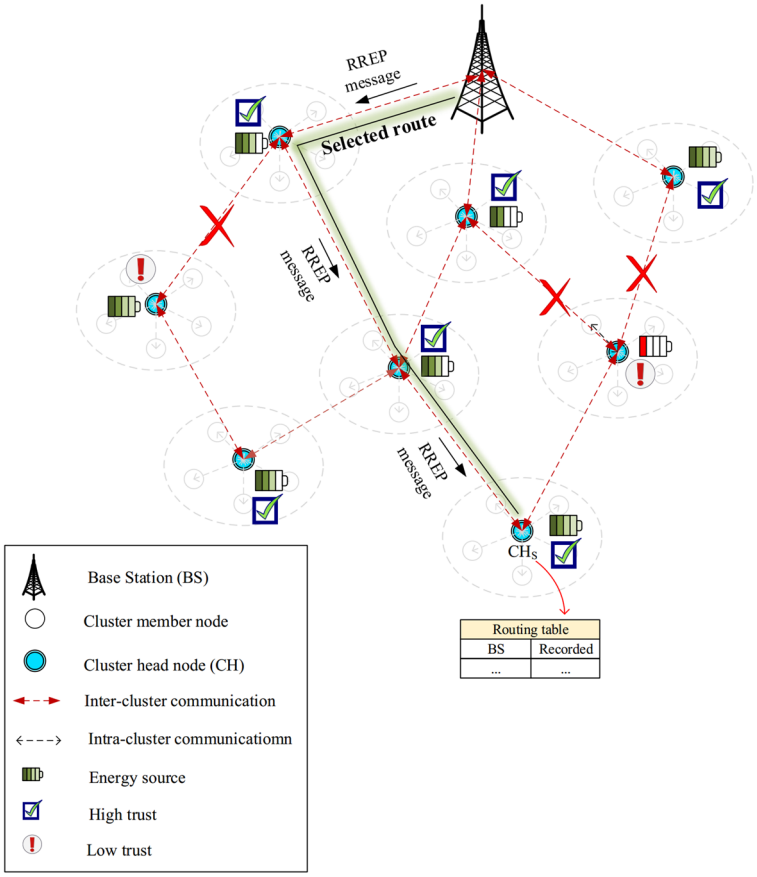

Figure 10. Sending the RREP message.

Parameter	Value
Simulation tool	NS2
The dimensions of network	$100 \times 100\,m^2$
BS position	(50, 100)
Total number of nodes	100
The number of nodes with 2 J energy	50
The number of nodes with 4 J energy	35
The number of nodes with 5 J energy	12
The number of nodes with 6 J energy	3
Transfer radius	20 m
Primary trust amount	0.5
Packet size	500 Bytes
The number of the data transfer operation in each round	100
E_{elec}	50 nJ/bit
ϵ_{fs}	10 pJ/bit/m^2
ϵ_{mp}	0.0013 pJ/bit/m^4
The primary population of fireflies	80
The number of iterations in FA	300 *times*

Table 3. Simulation settings.

energy efficiency and security. Once the RREQ reaches the base station, the RREQ broadcast process ends. See this process in Fig. 8.

Now, the base station must choose one path from the formed paths between CH_S and itself. For example, see ROUTE1 and ROUTE2 in Fig. 9. In the route selection operation, the base station utilizes the information recorded in RREQs to calculate the score of each path based on Eq. (27).

$$S_R = \frac{T_R}{E_R + D_R + H_c} \tag{27}$$

So that T_R, E_R, D_R, and H_c are the path trust (Eq. 27), the energy consumed in the path (Eq. 21), the delay taken in the path (Eq. 22), and hops counted in the path, respectively.

Then, the BS chooses the high-score path to transfer the data between CH_S and itself. Finally, the base station builds a route reply (RREP) message and transmits it for CH_S through the determined path. After receiving RREP, CH_S records the path in its table and uses it to transfer data to BS. See this process in Fig. 10.

Maintaining the formed paths. The path maintenance process is carried out to determine whether the formed path is cut (i.e. the route discovery process must be started again) or the formed path is connected. CHs regularly examine the connection status of the paths available in their routing table. For this reason, CH_S carries out the periodic transmission of a route validation message through the existing path. If the BS gets the message, it will confirm the connection of the path and transfers an acknowledgment (ACK) to CH_S. Otherwise, if CH_S does not receive any confirmation message from the BS for a certain period of time, CH_S is aware of the disconnection of the existing path and will begin the path search operation again in accordance with "Inter-cluster routing framework" section.

Algorithm 3 Inter-cluster routing process

Input: CH_i: Cluster head nodes ($i = 1, \ldots, N_{CH}$)
CH_S: Source CH
T_{CH_i}: Trust value of CH_i
T_i: Trust value of n_i
Output: Create a path between CH_S and BS.
Begin
1: if CH_S wants to send a data packet to BS then
2: CH_S: Searches in its routing table to find a valid path;
3: if there is a valid path between CH_S and BS then
4: CH_S: Send its data packet to BS through this path;
5: else
6: CH_S: Generate a route request (RREQ) message;
7: CH_S: Broadcast the RREQ message to its neighboring nodes such as CH_i;
8: while the RREQ message reaches BS do
9: if $E_{CH_i} < E_{threshold}$ or $T_{CH_i} < T_{threshold}$ then
10: CH_i: Remove the RREQ message;
11: else
12: CH_i: Broadcast the RREQ message to its neighbors;
13: end if
14: end while
15: BS: Select the best path among the discovered paths based on Equation 26;
16: BS: Unicast a route reply (RREP) message to CH_S;
17: CH_S: Record this route in the routing table;
18: CH_S: Send its data packet to BS through this path;
19: end if
20: end if
21: CH_S: Check the validity of the paths recorded in its routing table periodically;
22: CH_S: Send the route validation message through the available paths;
23: if CH_S does not receive any ACK message from destination then
24: CH_S: Remove the path from its routing table;
25: go to Line 1;
26: end if
End

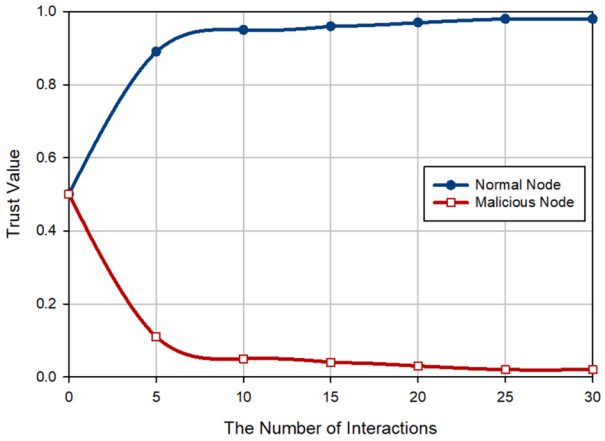

Figure 11. The performance measurement of FSRF based the trust evaluation.

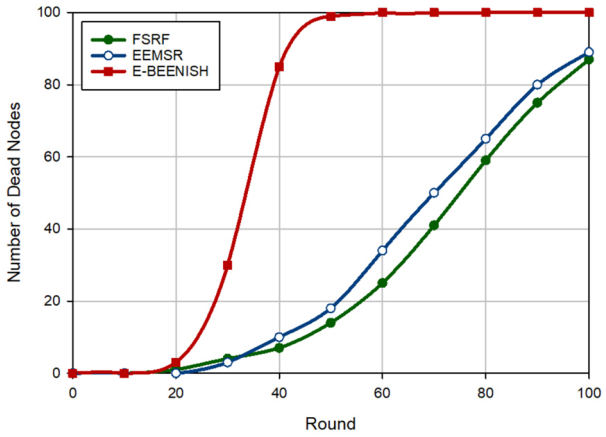

Figure 12. Evaluation of network longevity.

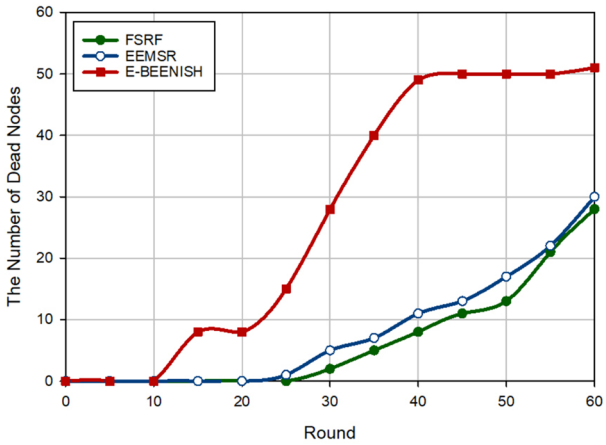

Figure 13. Evaluation of network longevity by changing the location of the BS to the corner of the network.

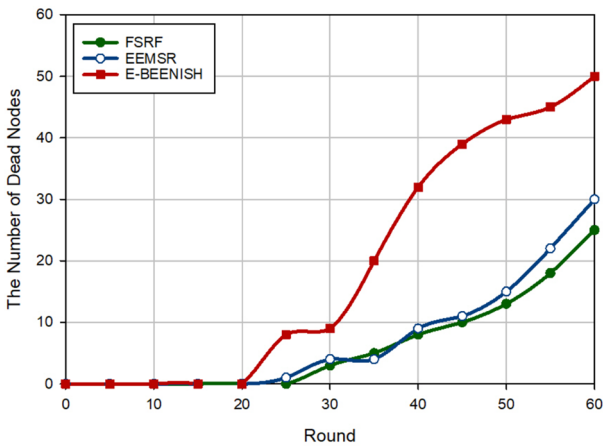

Figure 14. Evaluation of network longevity by changing the location of the BS to the center of the network.

Simulation and result evaluation

To accurately analyze the performance of FSRF, it must be carefully simulated and evaluated in different scenarios. For reaching this goal, the simulation operation is run in Network Simulator 2 (NS2) in accordance with the parameters listed in Table 3. According to the information recorded in this table, it can be found that

the dimensions of the simulation environment are $100 \times 100\,m^2$. In this process, 100 IoT nodes are randomly deployed in the simulation environment. These IoT nodes are immobile. They have heterogeneous energy sources, so there are 50 nodes with 2*J* energy, 35 nodes with 4 J energy, 12 nodes with 5 J energy, and 3 nodes with 6 J energy in the network. The transfer radius of these nodes is 20 meters. In FSRF, it is assumed that 10% of the nodes are hostile that are randomly selected from the network nodes with different energy levels. Each round includes 100 data transfer operations and each packet is 500 bytes. In this section, five test criteria are considered:

- *Criterion 1) Trust status:* This criterion evaluates the trust amount of the network nodes, whether hostile or normal, after various rounds and the exchange of information between the network nodes.
- *Criterion 2) Network longevity:* This criterion is used to analyze the lifetime of the network by counting the number of dead nodes in the network after various rounds.
- *Criterion 3) Energy level evaluation:* This criterion is used to measure the energy stored in the nodes after various rounds.
- *Criterion 4) Energy balance:* This criterion is used to evaluate whether the consumed energy is distributed between the network nodes evenly. To achieve this goal, the standard deviation of the energy consumed in nodes (SD_{Energy}) is calculated. If SD_{Energy} is close to zero, it confirms the balanced consumed energy between network nodes. However, if SD_{Energy} is close to one, it shows an imbalance energy consumption in the network.
- *Criterion 5) Packet delivery rate:* This criterion is for measuring the total number of data packets received at the destination compared to all packets sent from CH_S.

We compare FSRF to EEMSR and E-BEENISH. The selection of these two methods has several reasons:

- FSRF, EEMSR, and E-BEENISH are hierarchical and use clustering techniques to enhance energy efficiency in the network.

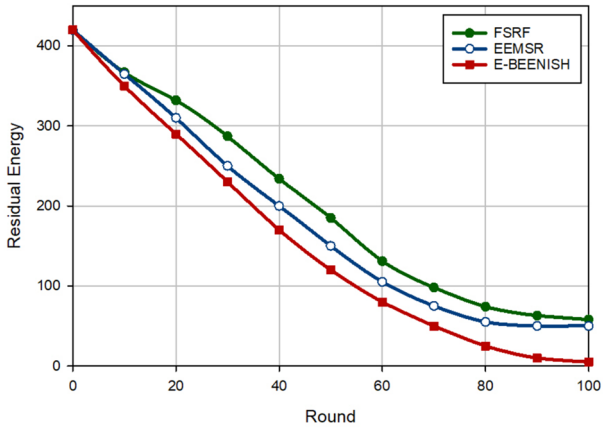

Figure 15. Evaluation of energy stored in nodes.

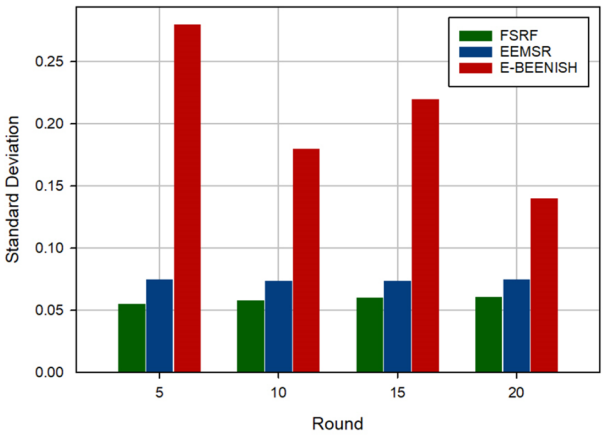

Figure 16. Evaluation of energy balance.

- EEMSR and FSRF have used metaheuristic algorithms to rise the performance of the IoT network so that EEMSR employs a genetic algorithm to correct the routing framework and FSRF employs the firefly algorithm to enhance the clustering process.
- EEMSR and FSRF have provided powerful trust mechanisms to protect the network nodes. However, E-BEENISH has not considered any security mechanism.

Trust amount. The first criterion for examining the performance of FSRF is to evaluate the trust amount of the network nodes. The results of this evaluation are displayed in Fig. 11. Note that there are two hypotheses in the performance measurement process: (1) The invading nodes are present in the network (i.e. 10% of the total network nodes) and (2) the initial trust of the nodes is adjusted to 0.5. Figure 11 shows when launching the network, it is difficult to distinguish the hostile nodes from the honest nodes because the exchange of information between the nodes is low and their trust is not well known. After increasing the exchange between the nodes, the fuzzy trust system designed in FSRF can help the nodes to be accurately aware of the trust status of themselves and their neighboring nodes. This increases the trust of the honest nodes to one and reduces the trust of hostile nodes to zero. In this case, FSRF can well separate hostile nodes from honest nodes.

Network longevity. The second criterion for examining the performance of FSRF is network longevity, which is obtained by counting the number of dead nodes at each round. In Fig. 12, FSRF has achieved the best network longevity compared to EEMSR (approximately 10.34%) and E-BEENISH (approximately 56.35%). Note that the performance of EEMSR and FSRF is very close to each other in terms of network longevity. If network longevity is defined based on the first node die (FND), EEMSR is superior to FSRF. However, if network longevity is defined based on half of the nodes die (HND) or the last node die (LND), the performance of FSRF

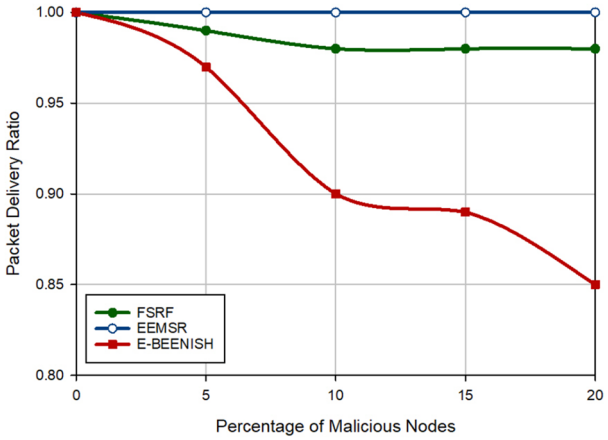

Figure 17. Evaluation of packet delivery rate.

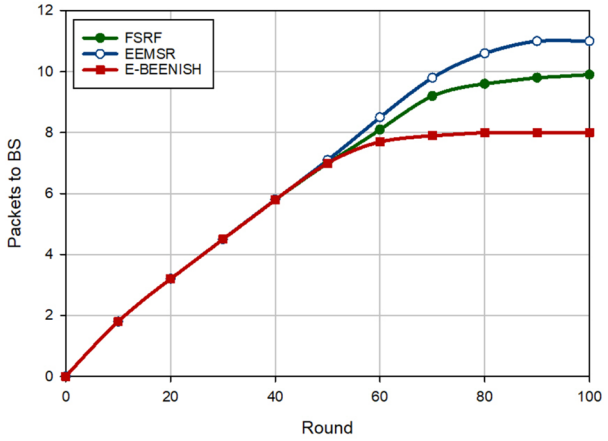

Figure 18. Evaluation of packets received by the base station.

is better than that of EEMSR and E-BEENISH. Now, this experiment is repeated by changing the location of the BS to analyze its effect on the routing schemes. In Fig. 13, the BS position is in the corner of the network, i.e. (0, 0). In this case, counting the number of dead nodes at each round indicates the improvement of this criterion by FSRF in comparison with EEMSR (16.93%) and E-BEENISH (74.79%). In Fig. 14, the BS is placed at the center of the network, i.e. (50, 50). In this figure, FSRF has succeeded in improving this criterion compared to EEMSR (14.50%) and E-BEENISH (66.65%). In these two experiments (Figs. 13 and 14), FSRF is superior to EEMSR in terms of FND. Additionally, the two experiments prove that changing the position of BS in FSRF and EEMSR cannot have a great impact on their performance. As a result, they are adaptable. However, this change in position has affected the performance of E-BEENISH. The better performance of FSRF in terms of network longevity is rooted in the attention to energy and security at the same time because FSRF only allows the trusted nodes with high energy to broadcast RREQ. Furthermore, FSRF has taken into account the amount of energy consumed in the discovered routes in the route selection process, but EEMSR does not pay attention to this parameter in the routing process. In addition, in the clustering process, the high-energy nodes are selected as CHs. E-BEENISH has not paid attention to the security of the network, so invaders can reduce network longevity by impacting the network performance. In E-BEENISH, each CH must send data packets to the BS in a one-hop way, which requires a lot of energy. However, EEMSR and FSRF have used intermediate nodes to send data to the base station. This improves energy balance in the network and thus extends network longevity.

Energy. The third criterion for examining the performance of FSRF is to evaluate the amount of energy stored in the network nodes. See Fig. 15. FSRF improves the energy stored in the nodes by 10.79% and 28.51%, compared to EEMSR and E-BEENISH, respectively. Given this figure, it can be said that FSRF and EEMSR have a good performance in terms of energy stored in the nodes. However, the E-BEENISH has a weaker performance in this criterion because E-BEENISH does not pay attention to the energy of the nodes in the CH selection process, as well as the direct transmission of data from each CH to BS has reduced the energy stored in the nodes. Lack of attention to network security can also reduce the energy stored in nodes due to the selection of untrusted CHs, the need to re-transfer data, and high packet loss. However, EEMSR and FSRF do not have these problems and consequently, show a better performance.

The fourth criterion for examining the performance of FSRF is to evaluate whether the energy is distributed between network nodes in a balanced manner. This criterion is determined by the standard deviation of the energy consumed in the nodes (SD_{Energy}). The results of this experiment are presented in Fig. 16. If SD_{Energy} is near zero, it confirms that the energy is consumed in a balanced manner. In contrast, if SD_{Energy} is close to one, it confirms that the energy is consumed in an imbalance manner. As Shown in Fig. 16, FSRF has the least SD_{Energy}, meaning that it can well balance the energy consumption between the network nodes. It reduced SD_{Energy} by 21.48% and about 71.46% compared to EEMSR and E-BEENISH, respectively.

Packet delivery rate. The last criterion for the evaluation of FSRF is to investigate the packet delivery rate on the network. The results of this evaluation are shown in Figs. 17 and 18. Figure 17 presents the results of PDR according to the change in the percentage of hostile nodes in the network. FSRF has about less PDR (about 1.4%) than EEMSR because the trust mechanism designed in EEMSR is more powerful than that in FSRF. Furthermore, FSRF improves PDR (approximately 6.94%) compared to E-BEENISH because E-BEENISH has not paid attention to network security. As a result, PDR in E-BEENISH drops rapidly by increasing the percentage of hostile nodes on the network. In Fig. 18, the number of data packets delivered to the destination (i.e. BS) has been examined. According to this figure, FSRF has a lower PDR (approximately 6.01%) than EEMSR. However, PDR in FSRF has improved by approximately 11.16% compared to E-BEENISH. These results prove that EEMSR is more powerful than FSRF in terms of security. Whereas, FSRF works better than EEMSR in terms of energy efficiency, which is stated in Figs. 15 and 16.

Conclusion

In this paper, a fuzzy secure hierarchical routing scheme based on the firefly algorithm (FSRF) was proposed for WSN-based IoT networks. This scheme seeks to achieve network security and energy efficiency. In FSRF, a fuzzy logic-based trust framework was presented to get the trust of nodes to detect and prevent various attacks such as black hole, flooding, wormhole, sinkhole, and Grey hole. Moreover, in FSRF, a FA-based clustering framework was designed to improve the energy consumption of nodes and network longevity. It comprises an objective function that considers trust amount, remaining energy, hops to BS, communication radius, and centrality. Finally, FSRF designs an inter-cluster routing framework to find reliable and energy-efficient paths on the network. Comparison of FSRF with EEMSR and E-BEENISH proved that the proposed method guarantees energy efficiency in the network because it improved network longevity by 10.34% and 56.35% and the energy stored in the nodes by 10.79% and 28.51% compared to EEMSR and E-BEENISH, respectively. However, FSRF is weaker than EEMSR in terms of security and has less PDR (almost 1.4%) than EEMSR. In future research directions, FSRF is evaluated under more scenarios to show its benefits and disadvantages. Moreover, we can test the robustness and efficiency of FSRF against various attacks. Furthermore, we will utilize new strategies like Q-learning and artificial neural networks (ANNs) to design robust trust frameworks to better separate abnormal nodes from normal nodes. For future work, this scheme can be improved for IoT networks with mobile nodes.

Data availability

All data generated or analyzed during this study are included in this published article.

Received: 14 March 2023; Accepted: 5 July 2023
Published online: 08 July 2023

References

1. Karlsson, J. Dooley, L. S. & Pulkkis G. Secure routing for MANET connected Internet of Things systems. In *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, 114–119. https://doi.org/10.1016/j.jnca.2016.03.006 (IEEE, 2018).
2. Yarinezhad, R. & Azizi, S. An energy-efficient routing protocol for the Internet of Things networks based on geographical location and link quality. *Comput. Netw.* **5**(193), 108116. https://doi.org/10.1016/j.comnet.2021.108116 (2021).
3. Boudagdigue, C., Benslimane, A., Kobbane, A. & Liu, J. Trust management in industrial Internet of Things. *IEEE Trans. Inf. Forensics Security* **25**(15), 3667–82. https://doi.org/10.1109/TIFS.2020.2997179 (2020).
4. Gali, S. & Nidumolu, V. An intelligent trust sensing scheme with metaheuristic based secure routing protocol for Internet of Things. *Cluster Comput.* **25**(3), 1779–89. https://doi.org/10.1007/s10586-021-03473-3 (2022).
5. Aruna, M. Ananda Kumar, S. Arthi, B. & Ghosh, U. Smart security for industrial and healthcare IoT applications. In Intelligent Internet of Things for Healthcare and Industry, 353–371 https://doi.org/10.1007/978-3-030-81473-1_17 (Springer, 2022).
6. Pal, S. & Jadidi, Z. Analysis of security issues and countermeasures for the industrial internet of things. *Appl. Sci.* **11**(20), 9393. https://doi.org/10.3390/app11209393 (2021).
7. Bhuiyan, M. N., Rahman, M. M., Billah, M. M. & Saha, D. Internet of things (IoT): A review of its enabling technologies in healthcare applications, standards protocols, security, and market opportunities. *IEEE Internet of Things J.* **8**(13), 10474–10498. https://doi.org/10.1109/JIOT.2021.3062630 (2021).
8. Muzammal, S. M., Murugesan, R. K., Jhanjhi, N. Z., Hossain, M. S. & Yassine, A. Trust and mobility-based protocol for secure routing in Internet of Things. *Sensors* **22**(16), 6215. https://doi.org/10.3390/s22166215 (2022).
9. He, D., Ye, R., Chan, S., Guizani, M. & Xu, Y. Privacy in the internet of things for smart healthcare. *IEEE Commun. Mag.* **56**(4), 38–44. https://doi.org/10.1109/MCOM.2018.1700809 (2018).
10. Yousefpoor, M. S. *et al.* Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review. *J. Netw. Comput. Appl.* **190**, 103118. https://doi.org/10.1016/j.jnca.2021.103118 (2021).
11. Gali, S. & Venkatram, N. Energy-Efficient cluster-based trust-aware routing for Internet of Things. In *Expert Clouds and Applications*, 493–509. https://doi.org/10.1007/978-981-16-2126-0_40 (Springer, 2022).
12. Farooq, U., Tariq, N., Asim, M., Baker, T. & Al-Shamma'a, A. Machine learning and the Internet of Things security: Solutions and open challenges. *J. Parallel Distrib. Comput.* **162**, 89–104. https://doi.org/10.1016/j.jpdc.2022.01.015 (2022).
13. Behura, A. & Priyadarshini, S. B. B. Application of the Internet of Things (IoT) for biomedical peregrination and smart healthcare. In *The Role of the Internet of Things (Iot) in Biomedical Engineering* 31–68. https://doi.org/10.1201/9781003180470-2 (Apple Academic Press, 2022).
14. Jeong, H. *et al.* SecAODV: A secure healthcare routing scheme based on hybrid cryptography in wireless body sensor networks. *Front. Med.* https://doi.org/10.3389/fmed.2022.829055 (2022).
15. Bedi, P. Goyal, S. B. Kumar, J. & Patnaik, P. Machine Learning Aspects for Trustworthy Internet of Healthcare Things. Internet of Healthcare Things: Machine Learning for Security and Privacy, 65–94. https://doi.org/10.1002/9781119792468.ch4 (2022).
16. Kanchana, V., Nath, S. & Singh, M. K. A study of internet of things oriented smart medical systems. *Mater. Today Proc.* **51**, 961–964. https://doi.org/10.1016/j.matpr.2021.06.363 (2022).
17. Alshamrani, M. IoT and artificial intelligence implementations for remote healthcare monitoring systems: A survey. *J. King Saud Univ.-Comput. Inf. Sci.* **34**(8), 4687–4701. https://doi.org/10.1016/j.jksuci.2021.06.005 (2022).
18. Hosseinzadeh, M. *et al.* A cluster-tree-based secure routing protocol using dragonfly algorithm (DA) in the Internet of Things (IoT) for smart agriculture. *Mathematics* **11**(1), 80. https://doi.org/10.3390/math11010080 (2023).
19. Yarinezhad, R. & Sabaei, M. An optimal cluster-based routing algorithm for lifetime maximization of Internet of Things. *J. Parallel Distrib. Comput.* **156**, 7–24. https://doi.org/10.1016/j.jpdc.2021.05.005 (2021).
20. Omolara, A. E. *et al.* The internet of things security: A survey encompassing unexplored areas and new insights. *Comput. Security* **112**, 102494. https://doi.org/10.1016/j.cose.2021.102494 (2022).
21. Adil, M. *et al.* HOPCTP: A robust channel categorization data preservation scheme for industrial healthcare internet of things. *IEEE Trans. Ind. Inform.* **18**(10), 7151–7161. https://doi.org/10.1109/TII.2022.3148287 (2022).
22. Muzammal, S. M., Murugesan, R. K. & Jhanjhi, N. Z. A comprehensive review on secure routing in internet of things: Mitigation methods and trust-based approaches. *IEEE Internet Things J.* **8**(6), 4186–4210. https://doi.org/10.1109/JIOT.2020.3031162 (2020).
23. Hatzivasilis, G., Papaefstathiou, I. & Manifavas, C. SCOTRES: Secure routing for IoT and CPS. *IEEE Internet Things J.* **4**(6), 2129–2141. https://doi.org/10.1109/JIOT.2017.2752801 (2017).
24. Mosenia, A. & Jha, N. K. A comprehensive study of security of internet-of-things. *IEEE Trans. Emerg. Top. Comput.* **5**(4), 586–602. https://doi.org/10.1109/TETC.2016.2606384 (2016).
25. Cao, J., Wang, X., Huang, M., Yi, B. & He, Q. A security-driven network architecture for routing in industrial Internet of Things. *Trans. Emerg. Telecommun. Technol.* **32**(4), e4216. https://doi.org/10.1002/ett.4216 (2021).
26. Nayagi, D. S., Sivasankari, G. G., Ravi, V., Venugopal, K. R. & Sennan, S. REERS: Reliable and energy-efficient route selection algorithm for heterogeneous Internet of things applications. *Int. J. Commun. Syst.* **34**(13), e4900. https://doi.org/10.1002/dac.4900 (2021).
27. Abbas, S. *et al.* Securing genetic algorithm enabled SDN routing for blockchain based Internet of Things. *IEEE Access* **9**, 139739–139754. https://doi.org/10.1109/ACCESS.2021.3118948 (2021).
28. Jazebi, S. J. & Ghaffari, A. RISA: Routing scheme for Internet of Things using shuffled frog leaping optimization algorithm. *J. Ambient Intell. Human. Comput.* **11**(10), 4273–4283. https://doi.org/10.1007/s12652-020-01708-6 (2020).
29. Seyfollahi, A. & Ghaffari, A. Reliable data dissemination for the Internet of Things using Harris hawks optimization. *Peer-to-Peer Netw. Appl.* **13**(6), 1886–1902. https://doi.org/10.1007/s12083-020-00933-2 (2020).
30. Zhang, Y. *et al.* An energy efficient multi-level secure routing protocol in IoT networks. *IEEE Internet Things J.*https://doi.org/10.1109/JIOT.2021.3121529 *(2021).*
31. Zhang, Y., Zhang, X., Ning, S., Gao, J. & Liu, Y. Energy-efficient multilevel heterogeneous routing protocol for wireless sensor networks. *IEEE Access* **7**, 55873–55884. https://doi.org/10.1109/ACCESS.2019.2900742 (2019).
32. Yang, X. S. Nature-inspired optimization algorithms: Challenges and open problems. *J. Comput. Sci.* **46**, 101104. https://doi.org/10.1016/j.jocs.2020.101104 (2020).
33. Yang, X. S., Deb, S., Fong, S., He, X. & Zhao, Y. X. From swarm intelligence to metaheuristics: Nature-inspired optimization algorithms. *Computer* **49**(9), 52–59. https://doi.org/10.1109/MC.2016.292 (2016).
34. Yousefpoor, E., Barati, H. & Barati, A. A hierarchical secure data aggregation method using the dragonfly algorithm in wireless sensor networks. *Peer-to-Peer Netw. Appl.* **14**(4), 1917–1942. https://doi.org/10.1007/s12083-021-01116-3 (2021).
35. Yang, X. S. Firefly algorithm. *Nature-inspired Metaheuristic Algorithms* **20**, 79–90 (2008).
36. Zadeh, L. A. Soft computing and fuzzy logic. *IEEE Softw.* **11**(6), 48–56. https://doi.org/10.1109/52.329401 (1994).
37. Zadeh, L. A. Fuzzy logic=computing with words. *IEEE Trans. Fuzzy Syst.* **4**(2), 103–111. https://doi.org/10.1109/91.493904 (1996).

38. Kulkarni, R. V., Förster, A. & Venayagamoorthy, G. K. Computational intelligence in wireless sensor networks: A survey. *IEEE Commun. Surv. Tutor.* **13**(1), 68–96. https://doi.org/10.1109/SURV.2011.040310.00002 (2010).
39. Heinzelman, W. R. Chandrakasan, A. & Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 10. https://doi.org/10.1109/HICSS.2000.926982 (IEEE, 2000)

Acknowledgements
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
A.M.R., M.S.Y., and M.H.: initial conceptualization. S.A., J.Y.: experimental setup. J.L., S.M.: field testing J.L., O.H.A., S.M., J.Y., S.A., A.M.R., M.S.Y., and L.T.: wrote the main manuscript text. M.S.Y., and L.T.: prepared figures. M.S.Y., M.H., J.Y., S.M.: wrote the analysis section. A.M.R. and J.L.: reviewed the final manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to A.M.R. or L.T.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.