



EATMR: an energy-aware trust algorithm based the AODV protocol and multi-path routing approach in wireless sensor networks

Huaying Yin¹ · Hongmei Yang² · Saeid Shahmoradi³

Accepted: 29 March 2022 / Published online: 31 May 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022, corrected publication 2022

Abstract

Rapid developments in radio technology and processors have led to the emergence of small sensor nodes that provide communication over Wireless Sensor Networks (WSNs). The crucial issues in these networks are energy consumption management and reliable data exchange. Due to the limited resources of sensor nodes, WSNs become a vulnerable target against many security attacks. Thus, energy-aware trust-based techniques have become a powerful tool for detecting nodes' behavior and providing security solutions in WSN. Clustering-based routings are one of the most effective methods in increasing the WSN performance. In this paper, an Energy-Aware Trust algorithm based on the AODV protocol and Multi-path Routing approach (EATMR) is proposed to improve the security of WSNs. EATMR consists of two main phases: firstly, the nodes are clustered based on the Open-Source Development Model Algorithm (ODMA), and then in the second phase, clustering-based routing is applied. In this paper, the routing process follows the AODV protocol and multi-path routes approach with considering energy-aware trust. Here, the optimal and safe route is determined based on various parameters, namely energy, trust, hop-count, and distance. In this regard, we emphasize the evaluation of node trust using direct trust, indirect trust, and a multi-objective function. The simulation has been performed in MATLAB software in the presence of a Denial of Service (DoS) attack. The simulation results show that EATMR performs better than the state-of-the-art methods in terms of successfully detecting malicious nodes and enhancing network lifetime, energy consumption, and packet delivery ratio. As a conclusion, EATMR shows an average of 4.3 and 6.1% superiority over M-CSO and SQEER in different scenarios, respectively.

Keywords WSN · Trust · Energy-aware · AODV protocol · ODMA · Multi-path routing

1 Introduction

The wireless sensor networks (WSNs) consist of a large number of distributed sensor nodes and often a sink node (base

station) that interacts with the environment through sensing physical parameters [1]. WSNs are widely used in many fields such as intelligent transportation, smart cities, environmental monitoring, smart agriculture, and healthcare [2]. These networks have certain specifications and limitations which differentiate them from other networks. The location of the sensor nodes is not necessarily determined in advance. This feature might lead to distributing some of the nodes in dangerous or inaccessible environments. That is to say that WSN-based protocols and algorithms need to bear a self-organization capability [3, 4]. This means that the sensor nodes need to be able to organize themselves within a network and subsequently be able to control and manage themselves efficiently [4].

Nodes in WSN rely on batteries that are limited in terms of energy, rechargeability, and replaceability [4–6]. In addition, they have limitations such as storage, memory, processing, radio range and bandwidth. WSNs use radio signals

✉ Hongmei Yang
yhmei1977@126.com

Huaying Yin
yinhying@sina.com

Saeid Shahmoradi
saeedsh61@yahoo.com

¹ Department of Computer and Information Engineering, Hebei Petroleum University of Technology, Chengde 067000, Hebei, China

² Department of Mathematics and Physics, Hebei Petroleum University of Technology, Chengde 067000, Hebei, China

³ Technical Office Engineer of Khuzestan Water and Power Authority (KWPA), Khuzestan, Iran

to exchange data over the network, which highlights the importance of routing algorithms in energy consumption and network lifetime. Routing algorithms can specify the route for transmitting data packets from source to destination, where routing can be single-hop or multi-hop. Single-hop data transmission over a long distance consumes more energy than multi-hop data transmission over the same distance with shorter steps. Accordingly, data transmission is done based on clustering approaches through which only nodes with the Cluster Head (CH) role are responsible for exchanging data to the sink. Therefore, after the clustering is done, each node transmits data directly toward its CH and it. CHs can transfer data received from member nodes as a packet to a sink or a neighboring node [7]. So far, various clustering-based routing algorithms have been proposed in WSN [4–6], but it is important to establish a safe route before performing routing. One of the effective tools to create a safe route is to use trust measuring techniques.

Misbehavior in WSNs is perceived in many forms such as packet loss, data restructuring, sending spurious packets, creating fake nodes, and so on [8]. Therefore, the WSN must be able to quickly and accurately identify security breaches. Due to the inherent nature of sensor participation in WSN, malicious nodes change state well and attack to network resources. Therefore, how to select participating nodes in data transmission is very important to increase efficiency [9]. Trust models are a reliable tool to achieve this purpose. These models can help nodes to identify malicious behaviors as well as appropriate decision-making [10]. Due to the importance of trust in WSN and limitations such as energy, storage, memory and processing, the problem of measuring trust for nodes has been raised [8, 10]. Accordingly, the provision of scalable trust models based on these limitations is of particular importance. In this paper, a distributed trust model is proposed in which nodes use direct and indirect techniques to measure trust, so that each node stores only the trust values of its neighboring nodes. As a result, this way of measuring trust and distribution makes the model scalable.

Many factors such as distance and energy affect the clustering of nodes. The amount of trust can also be considered as an important factor [11]. Meanwhile, most algorithms do not consider the trust factor for clustering and CHs selection [12]. Some algorithms use techniques such as cryptography and authentication to provide security on WSNs. In general, these techniques have poor connectivity and high computational overhead, which complicates the network [10]. Therefore, the need to provide an energy-aware trust technique with minimum complexity and overhead is essential to improve WSN security. Accordingly, we have developed a centralized clustering method based on optimization techniques to improve the routing process. EATMR uses ODMA and a multi-objective function to clustering and CHs selection. In

this paper, the routing process is based on the AODV protocol and the multi-path routes approach. Here, the optimal and safe route is determined based on various parameters such as energy, trust, hop-count and distance. Due to this routing process and how to measure trust, the proposed method is named as EATMR (Energy-Aware Trust based on the Multi-path Routing).

The main contributions of this paper are summarized as follows:

- Selection of safe nodes for routing based on a hybrid trust model
- Using the ODMA for clustering and selecting CHs
- Development of AODV routing protocol based on multi-path routes technique

The rest of this paper is organized as follows: Sect. 2 provides an overview of the challenges of WSN, ODMA, AODV protocol, and the energy model. Section 3 is devoted to literature review. Details of the proposed EATMR scheme are described in Sect. 4. Section 5 describes the results of the simulations and comparisons, and finally, Sect. 6 summarizes the conclusion of the paper and presents future work.

2 Background

In this section, we first review the challenges of WSN. After that, the details of the AODV protocol that we use for routing are described. Finally, the energy consumption model used in this study is expressed.

2.1 Challenges in WSN

In general, there are many challenges in WSN, among which this paper mainly has focused on the trust problem. However, other challenges such as setup, clustering, CHs selection, and routing must be emphasized to make the network operational [1, 5]. Some of the most important challenges of WSN are shown in Fig. 1, where the challenges studied in this paper are identified. Challenges such as energy consumption and network lifetime are often considered as objectives and evaluation metrics, while improving other challenges such as clustering and routing can improve them [1].

In addition to the challenges discussed in this paper, there are other challenges in WSNs. One of the major problems that cause a collision in WSN is Hidden and Exposed terminal problem. This problem can be resolved by the Time Division Multiple Access (TDMA) protocol [11]. Bandwidth constraint can directly affect the exchange of messages between nodes. An appropriate node deployment scheme in WSNs can reduce the complexity of problems such as clustering and routing. Reliability refers to the secure transfer of data to the

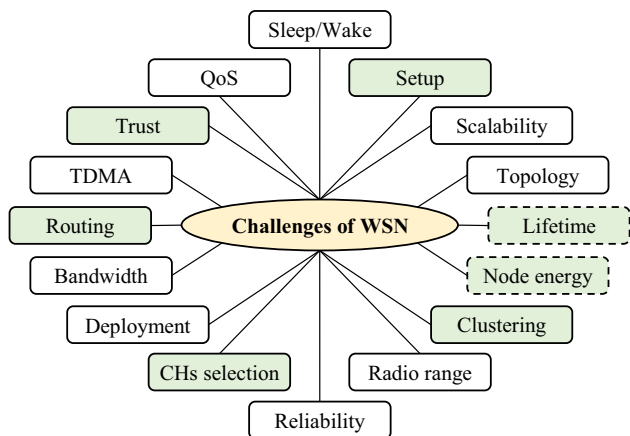


Fig. 1 Challenges of WSN along with the challenges examined

destination. The radio range will affect the number of neighboring nodes, which collaborate to forward data to the sink. Reportedly, topology can be effective in reducing energy consumption in WSNs [12]. Scalability refers to the deployment of nodes in the environment to maintain performance. Node sleep/wake scheduling is an essential consideration in WSNs that can minimize communication overhead and computation. A lot of research has been done in various areas of WSNs, but real-time communication with the Quality of Service (QoS) concept is still unexplored.

2.2 AODV protocol

One of the main routing protocols in WSNs is Ad-hoc On-Demand Distance Vector (AODV) [14]. In this protocol, each node has a routing table, so that in this table the routes of all nodes in the network along with the distance to them are stored. This protocol uses control packets of Route Request (RREQ), Route Response (RREP) and Route Error (RERR) to determine the appropriate route [14]. RREQ, RREP, and RERR represent the destination sequence number, hop-counts, and route failure, respectively. In general, routing in the AODV protocol involves two processes: route discovery and route maintenance.

In the route discovery process, the source node broadcasts the RREQ packet to its neighbors. Each of the neighboring nodes that has an active route between the source and the destination in its routing table, notifies the source by sending an RREP packet. Otherwise, each node broadcasts the RREQ packet to its neighbors. This process is repeated until the RREQ reaches the destination or an intermediate node of an active route to the destination with a sequence greater than or equal to the RREQ sequence. After completing the RREQ broadcast step, the RREP is sent from the destination node in the reverse-routes of the intermediate nodes to the source node. When a node loses connectivity to its next hop, the node

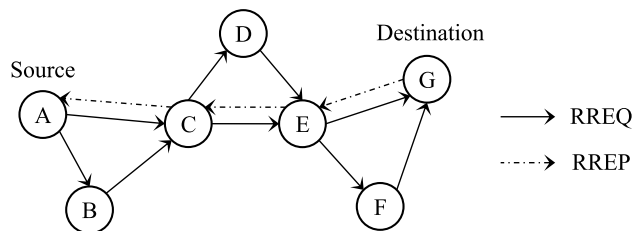


Fig. 2 Routing process in AODV protocol

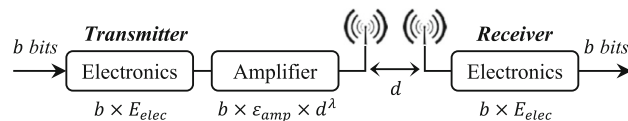


Fig. 3 First order communication mode

invalidates its route by sending an RERR to all nodes that potentially received its RREP. In the maintenance process, each node can inform its neighbors using a local broadcast, which is called Hello packets [14]. The routing process in the AODV protocol is as shown in Fig. 2.

2.3 Energy model

In this paper, the first order communication mode is used to energy management in the use of sensor nodes [1, 10]. In this model, the energy consumption for the transmitter and receiver nodes is defined according to Fig. 3.

A packet consisting of b bits is transmitted between the transmitter (T_x) and the receiver (R_x) at a distance of d meters based on the energy E_{T_x} . The transmission energy for the transmitter node is defined by Eq. (1).

$$E_{T_x} = b \times E_{elec} + b \times \epsilon_{amp} \times d^\lambda \tag{1}$$

where E_{elec} is the energy consumed by the transmitter circuitry for one bit and $b \times E_{elec}$ is the energy required by the transmitter to propagate a packet with b bits. ϵ_{amp} is the energy of the transmitter signal amplifier over the distance, and λ represents the route drop constant, so that $\lambda = 2$ is related to free space propagation model and $\lambda = 4$ is related to the multi-path fading propagation model.

The value of λ is determined depending on the transmission distance d relative to the threshold distance d_0 [1, 10], which is usually considered based on Eq. (2).

$$d_0 = \sqrt{\epsilon_{fs} / \epsilon_{mp}} \tag{2}$$

In addition, the energy required to receive b bits of data by the receiver are calculated according to Eq. (3).

$$E_{R_x} = b \times E_{elec} \tag{3}$$

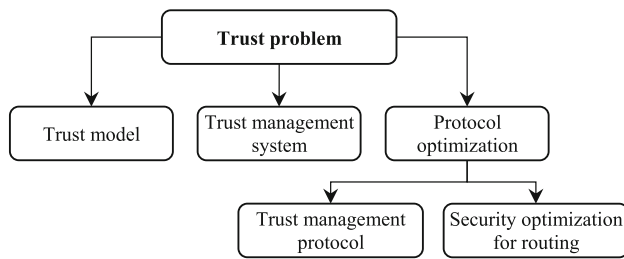


Fig. 4 Classification of trust-related works in WSN

Accordingly, the energy required to transfer data between nodes s_i and s_j (i.e., the transmission cost of the connection) is denoted by $e_{i,j}$, as shown in Eq. (4).

$$e_{i,j} = E_{T_x}(i) + E_{R_x}(j) \quad (4)$$

In this paper, the parameters of the energy model are adjusted according to [1, 10]. Therefore, the size of the data packets is $4KB$, the size of the Hello packets is $25B$, the initial energy of the nodes is $0.2J$, the energy required to sense a bit of data from the environment is $5 \times 10^{-9}J/B$, the energy required to aggregate, compress and build a packet for each bit is $5 \times 10^{-9}J/B$, and the energy required for a node to awake (changing the state from sleep to wake) is $2nJ$. In addition, E_{elec} is set to $50nJ/B$ and ε_{mp} to $100pJ/B/m^2$.

3 Literature review

In this section, the research literature related to the problem of trust in WSN is reviewed and then the limitations and research gap are discussed. The structure of this section is similar to approach as [15]. In general, studies on the trust in WSN are often summarized based on various fields such as trust model, trust management system, and protocol optimization [16]. A classification of trust-related works is provided in Fig. 4.

3.1 Trust model

The trust model provides a reliable communication management mechanism between nodes through which safe nodes can be trusted to participate in the routing process. Extensive research has been proposed in the literature to design trust models in the WSN, some of which are discussed below.

Gilbert et al. [17] developed a Time Series Trust Model (TSTM) based on Trust-based Auto Regressive (TAR) and Toeplitz matrix for WSNs [17]. The performance of TSTM has been proven to identify malicious nodes based on reconstruction and aggregation against three different attack types. In this model, data reconstruction is performed based on the Basis Pursuit algorithm, which provides the best performance

against bad-mouthing attack. Ghugar et al. [18] introduced a Layer trust-Based Intrusion Detection System (LB-IDS) to improve the security of WSNs [18]. This model uses the standard deviation of trust in each layer on attacks to measure trust. LB-IDS can defend against sinkhole attack in network layer, back-off manipulation attack in MAC layer and jamming attack in physical layer.

Zhao et al. [19] proposed an Exponential-based Trust and Reputation Evaluation System (ETRES) for WSN [19]. ETRES uses the exponential distribution and interactions of nodes in the network to measure the trust of nodes. In this system, the entropy method is used to estimate the uncertainty of direct trust scores. Also, indirect trust is measured when the uncertainty of direct trust is relatively high. In addition, ETRES updates trust scores at various rounds to reduce the detrimental effects of malicious nodes. Kalidoss et al. [20] developed the QEER (QoS aware Energy Efficient Routing) protocol and introduced the SQEER (Secured QoS aware Energy Efficient Routing) protocol [20]. QEER considers reliability over QoS and does not focus on security or latency. SQUIER provides reliability according to QoS, trust modeling and key-based authentication. In addition, SQEER performs routing based on the clustering technique and selects CHs based on security factors.

Wu et al. [21] proposed a Beta and LQI based Trust Model (BLTM) for WSN [21]. LQI is known as a link quality indicator and is used to stabilize the nodes trust with poor-quality links. Therefore, BLTM considers the adverse effect of poor-quality links on the trust score to measure direct trust. Here, direct trust is measured based on energy, communication, and data, and then the weight of each factor is discussed. Anwar et al. [22] proposed a Belief based Trust Evaluation Mechanism (BTEM) for WSN [22]. BTEM uses Bayesian belief to detect malicious nodes. Bayesian belief uses more of the data collected over time to measure direct and indirect trust. BTEM defends WSN well against DoS, On-Off and Bad-mouth attacks. Nie [23] presented a Trust model of Dynamic optimization based on entropy (Trust-Doe) for WSNs [23]. This model groups the nodes based on the degree of global trust. Then, it uses the entropy method to determine the weight of the node in each group. Trust-Doe can measure and update the local trust score of nodes using the group local evaluation standard deviation and entropy values. Although this model improves the ability to detect malicious nodes, it does not take into account energy consumption.

3.2 Trust management system

Trust management systems in WSNs use behaviors and interactions between nodes in the network to identify malicious nodes and measure distributed trust scores. The reason for using the distributed policy for measuring trust in such systems is the limitations of the sensor nodes. Therefore, these

systems have the advantages of scalability and flexibility in considering behaviors and interactions. There are several works in the literature that have presented trust management systems, some of which are discussed below.

Jinhui et al. [24] proposed an Intrusion Detection System based on Energy Trust (IDSET) for DoS combined attacks on the WSN [24]. The system can detect network intrusion using energy-aware trust and node energy analysis. The authors present an energy series correlation study based on the energy consumption prediction that can effectively reduce the impact of DoS combined attacks on network traffic. Firoozi et al. [25] proposed a hierarchical trust management model for distributed WSN in which the network area is divided into cells of equal size [25]. This scheme is known as DiSLIP (Distributed Subjective Logic-based In-network data Processing). This model generates reliable nodes based on interactions between nodes and considers temporal and spatial correlations. It also uses a subjective logic-based scheme to measure trust. In addition, this study proposes an energy saving mechanism to increase network reliability.

Janani and Manikandan [26] propose a public key infrastructure (PKI) model for mobile ad hoc networks (MANETs) [26]. This model is known as JANANI and uses the Bayesian theorem in hexagonally clustered MANET to measure the distributed trust score. Thus, JANANI provides an effective security approach based on distributed trust and hierarchical clustering for WSN. Sahoo et al. [27] proposed a lightweight trust management model based on punishment and reward policy called GATE [27]. GATE uses a dynamic time sliding window mechanism to counteract various attacks and measure the trust scores. GATE detects malicious nodes more quickly and requires fewer resources than similar schemes. The disadvantage of this scheme is the lack of use of recommendations to measure trust, which has led to a decrease in efficiency against bad-mouthing attacks.

3.3 Protocol optimization

There are many approaches in the literature of protocol optimization in WSN that focus mainly on the two fields of trust management protocol and security optimization for routing. Most routing protocols in WSNs are based on clustering, known as Hierarchical Routing Protocols (HRPs) [10]. HRP uses intermediate nodes and multi-hop routing instead of sending data directly to the sink. To do so, HRP forms clusters and transmits data through CHs. In this subsection, some new methods related to protocol optimization are discussed.

Patil et al. [28] used the Monarch-Cat Swarm Optimization (M-CSO) algorithm for routing work in WSN [28]. M-CSO provides a trust-based opportunistic routing framework using hybrid optimization. M-CSO is a combination of Monarch Butterfly Optimization (MBO) and Cat Swarm

Optimization (CSO). In this algorithm, first the safe nodes are selected based on the tolerant constant mechanism and then the opportunistic nodes are selected from the safe nodes. The tolerance constant is modeled based on the parameters of trust, connectivity and QoS. M-CSO provides the ability to detect and defend against DoS and Blackhole attacks by developing trustworthy and adaptive routing. Khan et al. [29] proposed an Energy-aware Trust-based Efficient Routing Scheme (ETERS) for WSNs [29]. ETERS is a realistic multi-trust, comprehensive, and scalable model for dealing with internal attacks on WSNs that emphasizes beta distribution strength and weighting methods. The scheme uses a flexible penalty coefficient to prevent attacks according to the needs of the network. ECHSA measures the trust score with a trust-based attack detection algorithm (TADA) algorithm based on the parameters of ID, triple trust and location. In addition, ECHSA proposes an efficient CH selection algorithm that can maintain load balance for routing.

Sun and Li [30] proposed a comprehensive trust-aware routing protocol for WSN that uses attributes such as energy, communication, data, and recommendation [30]. The scheme is called TRPM (Trust-aware Routing Protocol with Multi-attributes), which uses an improved sliding time window according to the frequency of attacks to identify attackers. The simulations show that TRPM increases the average packet delivery rate by about 19% compared to similar protocols. Wang et al. [31] proposed an Energy-efficient Trust Management and Routing Mechanism (ETMRM) for software-defined networking based WSN [31]. In ETMRM, the SensorFlow table is first developed to implement the trust monitoring and evaluation plan, and then malicious nodes are identified based on the measured trust scores. In addition, the authors proposed an efficient message aggregation scheme to reduce energy consumption and increase the reliability of data exchange. Mehetre et al. [32] proposed a Trustable and Secure Routing Scheme (TSRS) using a two-step security mechanism that aims to combat internal attacks in the WSN [32]. TSRS uses active initiative trust to ensure routing protocol and cuckoo search to identify reliable route. This scheme also ensures an increase in network lifetime.

3.4 Limitations and research gap

In general, the most highlighted limitation of trust-related studies in WSNs is the lack of hardware and complex configuration of models for simulating large networks (networks with a high number of nodes). In addition, this limitation is mentioned in most similar studies, and the authors perform simulations only for small networks. In this regard, according to the simulation results in this paper, it can be predicted that the results will be similar for large networks. If the network range is so large that the sink cannot communicate directly

with all nodes, then this can be done as multi-hops through the routing algorithm.

In general, WSN security research often focuses on areas such as defend against attacks, identify malicious nodes, and trust score calculation. Malicious nodes can be identified in a timely manner by optimally computing the trust score. There are many parameters for computing the trust score that have been used in various studies, as discussed in the previous three subsections. As a conclusion, not all parameters in all works are considered to measure trust. This can be due to the complexity of measuring trust in WSNs with many parameters. In addition, there are many internal attacks on WSNs that can be referred to as DoS, bad-mouthing, collusion, on-off, sinkhole, blackhole, conflicting behavior and data forgery [15], where all studies consider some of these attacks to measure trust. Therefore, it is not necessary to use all the available parameters to measure trust in WSNs.

However, we try to reduce the computational complexity and counter more attacks by considering some appropriate parameters for measuring trust. However, the simulation for the proposed scheme (i.e., EATMR) was performed only in the presence of a DoS attack, however, according to the parameters considered to measure trust, this scheme has the ability to defend against various attacks such as bad-mouthing, on-off, sinkhole and blackhole. Because, the proposed method calculates the trust score based on the behavior of the nodes in sending and receiving packets and does not depend on the type of attack, so it can be concluded that the results of other attacks can be similar to the DoS attack. Table 1 summarizes the differences between the EATMR and the literature review based on the trust measurement parameters, and Table 2 shows this comparison for a set of internal attacks that works can defend against.

4 EATMR scheme

Due to the complexity and high cost of WSN design, it is necessary to simulate and evaluate the network protocols before implementation. The purpose of the simulation is to discover new ideas faster under different conditions. So far, many routing protocols have been introduced to increase security in WSN [30–32]. In this paper, an Energy-Aware Trust-based Multipath Routing (EATMR) scheme is proposed to improve security in WSN. EATMR is expressed in four main phases: (1) topology configuration and network setup, (2) determine safe nodes, (3) clustering and CHs selection, and (4) clustering-based routing.

Safe nodes are determined in the second phase, which uses energy, connectivity and trust parameters. The third phase is related to the clustering and CHs selection, for which we use ODMA. In general, clustering is an effective way to reduce

energy consumption in WSN that can extend network lifetime. In addition, nodes in WSN are susceptible to many security attacks, and the safe selection of CHs can improve WSN security. Hence, CHs in EATMR are selected based on a safety level. Finally, clustering-based routing is performed in the fourth phase, where only secure nodes will be involved in routing. The EATMR is based on an energy-aware trust routing algorithm that uses the AODV protocol and the multi-path routes approach. The proposed routing algorithm analyzes all the routes detected by AODV based on a hybrid fitness function and then selects the optimal route to send data to the sink. The EATMR architecture is shown in Fig. 5. For the convenience of the reader, all the math notations related to this paper are provided in Table 3.

4.1 Network setup

This study is designed in a small simulation environment in the range of $M \times M$ meters. The network topology consists of three types of nodes: normal node, malicious node and sink node. Normal nodes sense environment data and transfer it to the sink. Malicious nodes are also a type of normal node that has been attacked by DoS [16].

From a topological point of view, the network consists of N nodes whose node i is represented by s_i . The placement of nodes in the environment is done randomly with a uniform distribution and the locations are fixed until the end of the simulation. From a communication point of view, each node has a limited radio range. The radio range of all nodes is the same and is determined by the distance d_0 . Therefore, the power of data exchange between two nodes is limited to d_0 . Sensor nodes are location-aware, i.e., equipped with a global positioning system (GPS). Therefore, the sink can calculate the distance between nodes with the Euclidean relation [1], which is used here to show the distance between nodes s_i and s_j of $d_{i,j}$. Also, each node such as s_i has limited energy which is indicated by e_i . Meanwhile, the sink has an unlimited amount of energy [20–23].

Other hypotheses considered in the proposed scheme are as follows:

- The behavior of the nodes is analyzed based on the packets sent and received between them.
- Malicious nodes cannot communicate with each other. This means that the attack details of malicious nodes in the network cannot be shared between other malicious nodes.
- The size of all Hello packets is the same, and this hypothesis also exists for data packets.
- The sensed information is the same in all nodes and the aggregation function in CHs is calculated based on the average of the values .

Table 1 Comparison between our work and existing works for the trust measurement problem in WSNs

Trust schemes	Parameters							
	Interaction	Behavior	Connectivity	Energy	QoS	Traffic	Latency	Authentication
TSTM	✓	✓	×	×	✓	×	✓	×
LB-IDS	×	✓	✓	×	×	×	×	×
ETRES	×	✓	×	✓	×	✓	✓	×
SQEER	✓	✓	×	×	✓	×	×	✓
BLTM	✓	✓	✓	✓	×	✓	×	×
BTEM	✓	✓	×	×	×	✓	✓	×
Trust-Doe	×	✓	✓	×	×	×	✓	×
IDSET	✓	×	×	✓	×	✓	×	×
DiSLIP	✓	✓	×	✓	✓	×	✓	×
JANANI	✓	×	✓	×	×	×	×	✓
GATE	×	✓	×	×	✓	✓	✓	×
M-CSO	✓	×	✓	×	✓	×	×	×
ETERS	✓	✓	✓	✓	×	×	×	×
TRPM	✓	✓	✓	✓	✓	×	×	×
ETMRM	✓	×	×	✓	✓	×	×	×
TSRS	✓	✓	✓	×	×	✓	×	×
EATMR	✓	✓	✓	✓	✓	×	✓	×

‘✓’ indicates the use of a parameter to measure the trust score and ‘×’ indicates not to use that parameter

- Each node sends data to its CH according to the TDMA (Time-Division Multiple Access) scheduling and sleeps until the beginning of the next timeslot.

After the nodes are completely deployed, the sink clusters the nodes and selects the appropriate CHs. It then notifies each node of its role by sending Hello packets. This is repeated by the sink if there is a change in clustering and CHs. In certain timeslots, each node sends the remaining energy and trust of its neighbors to its CH, and the CH also sends them to the sink. In order to reduce the cost of sending this information to the sink as well as the design of the distributed trust model, this timeslot is considered relatively large. Therefore, the sink is aware of the trust and energy of all nodes and can use it to cluster nodes and select CHs. Each node, in addition to its internal memory, has a neighbor table and a routing table. In the memory of each node, in addition to the remaining energy, the parameters needed to measure the trust are also stored, where these parameters are sent to the nodes by the sink.

The trust score is calculated based on the behavior of the nodes when receiving and sending packets. Therefore, this score is not fixed and should be updated in the identified timeslots. Meanwhile, according to the distributed model for calculating the trust score, nodes interact only with neighboring nodes and due to the small distance between nodes, little energy is consumed.

Each node in certain timeslots measures the direct trust score of all its neighbors. It then sends the trust scores along with its residual energy to all neighboring nodes. Each node creates/updates its own neighbor table by receiving a Hello packet from a neighboring node. An example of a neighbor table structure is shown in Fig. 6. In this table, ‘NodeID’ contains the ID of the neighboring nodes, and each neighbor has an ‘ExpireTime’. ‘Energy’ refers to the amount of energy remaining in the neighboring node. ‘Trust’ and ‘Recommended trust’ refer to the trust score and the recommended trust score for each neighbor, respectively. ‘Safe’ with a binary value indicates the reliability of neighboring nodes, where nodes with a value of 1 are reliable and can participate in routing. The process of determining safe nodes is described in the next section.

Trust fields are used to estimate indirect trust. Since not all neighbors of the two nodes are common, each node only stores its neighbors’ information in the routing table and does not consider the information received for the other nodes. In addition, due to the existence of common neighbor nodes, a node may receive more than one trust score for some of its neighbors. In this case, the trust score for these nodes is measured as the average. For example, let G be a complete graph with a set of nodes $\{a, b, c, d\}$. For a node such as d , the data received from nodes a, b , and c are $a = [e_a = 0.1, T_b = 0.3, T_c = 0.5, T_d = 0.4]$, $b = [e_b = 0.15, T_a = 0.5, T_c = 1, T_d = 0.3]$, and

Table 2 Comparison between our work and existing works based on the type of attacks detectable in WSNs

Trust schemes	Parameters									
	DoS	Bad-mouthing	Collusion	On-off	Sinkhole	Blackhole	Conflicting behavior	Data forgery		
TSTM	✓	✓	×	×	×	×	×	✓		
LB-IDS	×	-	×	+	✓	-	×	-		
ETRES	×	×	✓	✓	×	×	-	-		
SQEER	✓	×	-	-	-	✓	×	-		
BLTM	✓	×	×	-	-	+	-	✓		
BTEM	✓	+	-	✓	-	-	-	×		
Trust-Doe	×	-	✓	×	×	×	×	-		
IDSET	✓	-	-	-	-	×	-	×		
DiSLIP	-	✓	-	-	-	-	×	-		
JANANI	-	×	×	×	✓	✓	-	-		
GATE	×	✓	×	✓	×	-	×	×		
M-CSO	✓	+	-	-	×	✓	×	-		
ETERS	+	×	×	-	✓	✓	-	-		
TRPM	+	×	-	-	✓	✓	+	+		
ETMRM	×	-	×	×	✓	✓	×	-		
TSRS	×	×	×	×	×	✓	×	×		
EATMR	✓	✓	+	✓	✓	✓	+	-		

Note: '✓' indicates the ability to defend against attack. ' + ' indicates the ability to detect an attack. ' - ' indicates that it is unknown to the attack and ' × ' indicates inability to detect and defend against the attacks

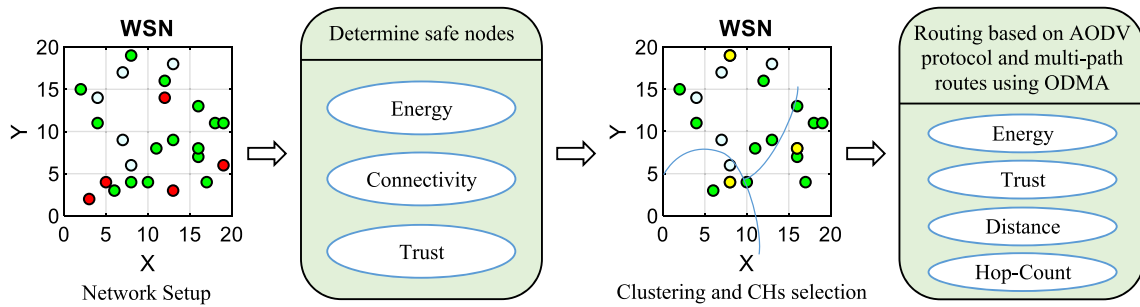


Fig. 5 EATMR architecture

Table 3 Description of the math notations

Symbols	Description	Symbols	Description
T_x	Packet transmitter	e_i	Remaining energy of i -th node
R_x	Packet receiver	$e_{i,j}$	Energy required to transfer data between s_i and s_j
E_{T_x}	Energy required to transmission	$d_{i,j}$	Distance between s_i and s_j
E_{R_x}	Energy required to receive	w	Weight coefficient for clustering
d_0	Threshold distance	K	Total number of clusters
E_{elec}	Energy consumed by the transmitter circuitry	D_v	Sum of the intra-cluster distance
ε_{amp}	Energy of the transmitter signal amplifier	σ_e	Standard deviation of the energy of the clusters
λ	Route drop constant	σ_c	Standard deviation of the number of candidate nodes from all clusters
s_i	i -th node	D_C	Average distance of all candidate nodes to the sink
DT_j	Direct trust score	E_r	Sum of the energies of the nodes participating in route r
IT_j	Indirect trust score	T_r	Sum of the trust scores of the nodes participating in route r
α	Impact coefficient for trust scores	D_r	Sum of the distances of the nodes participating in route r
ar_j	Number of acknowledgement packets received by s_j	HC_r	Hop-count of the nodes participating in route r
nr_j	Total number of packets received by s_j	ξ	Weight coefficient for routing
at_j	Number of acknowledgement packets transferred by s_j	E_j	energy rate of s_j
nt_j	Total number of packets transferred by s_j	C_j	Connectivity of s_j
β	Impact coefficient between packets sent and received	T_j	Trust score of s_j
nn_j	set of neighboring nodes of s_j	E_0	Initial energy
T_k^j	Recommended trust to s_j by s_k	e_{max}	The node with the most energy
θ_{SN}	Threshold for updating safe nodes	nc_j	Number of links of the j -th node
θ_{MN}	Threshold for identifying safe nodes	L	Total number of connections in the network
θ_{CA}	Threshold for updating clustering	N_P	Population size in ODMA
θ_{CH}	Threshold for updating CHs	z	Number of leading software
SV_j	Safety level of node j	ρ	Probability of promising software change
R_{MN}	Rate of malicious nodes	s_{old}	Latest position of the software
N	Number of sensor nodes	s_{cur}	Current position of the software
M	Range of simulation environment	s_{new}	New position of the software

NodeID	Energy	Trust	Recommended trust	Safe	ExpireTime
ID _a	0.10	0.60	0.40	1	10
ID _b	0.15	0.35	0.30	0	10
ID _c	0.05	0.75	0.60	0	10

Fig. 6 Neighbors table structure

$c = [e_c = 0.05, T_a = 0.7, T_b = 0.4, T_d = 0.6]$, respectively, where e refers to energy and T refers to trust. Since the data is sent to node d , T_d is the recommended trust score to d by the sender node. Accordingly, the neighbor table of node d is based on Fig. 6. Here, the expire time for the neighborhood is 10 rounds.

Due to the use of AODV protocol for routing, routing tables are managed according to the policy of this protocol [33]. The AODV protocol needs to have the following information inside each of the routing table inputs: NodeID destination, destination sequence number, hop-count to destination, NodeID neighboring nodes for next-hop in route, route validity period, a list of other neighbors participating in this route, a buffer to ensure reviewing the route requests. In this paper, in addition to these fields, connectivity and distance are also stored in the routing table. We will use connectivity to compute the safety level of the nodes and distance to determine the optimal route. Based on these fields, the route between nodes is searched by the AODV protocol. A route is retained in the routing table as long as it is required during a route maintenance procedure. Therefore, routing tables are created dynamically on demand.

Routing may be required to determine safe nodes or perform clustering steps. In this paper, before creating routing tables based on defined rules, routing is performed based on the classic AODV protocol. In other words, the minimum routing requirements are met by the AODV protocol when setting up the network. In general, network setup and the hypotheses defined in this section are presented according to some similar studies such as [11] and [12].

4.2 Identifying safe nodes

In EATMR, the safe nodes are determined based on the parameters of energy, connectivity and trust score, and the routing process is performed based on them. Energy and connectivity are available through neighbor and routing tables, respectively, and the trust score is measured as distributed by each node. Safe nodes are detected/updated at specific timeslots (for example, each θ_{SN} routing round). Let SV_j be the safety level of node j . The trust score for all nodes is initially set to 1, so all nodes are safe in the first round. However, after analyzing the behavior of the nodes, the energy and the trust score for the nodes change and the safe nodes must be determined with a threshold. The EATMR uses the

θ_{MN} threshold to isolate safe and malicious nodes. Therefore, the ‘Safe’ field of the routing table is calculated based on Eq. (5).

$$Safe_j = \begin{cases} 0 & SV_j < \theta_{MN} \\ 1 & otherwise \end{cases} \quad (5)$$

where SV_j is computed based on the parameters of energy, connectivity and trust score, as shown in Eq. (6). Due to the difference between the type of parameters, the scale of all parameters is normalized between 0 and 1. In addition, each node is not allowed to compute its own safety level and can only compute the safety level of its neighboring nodes. Therefore, SV_j is computed by the neighboring nodes of s_j .

$$SV_j = \frac{1}{3}[E_j + C_j + T_j] \quad (6)$$

where E_j , C_j and T_j are the energy rate, connectivity and trust score of s_j , respectively, which are discussed below.

Energy rate In most attacks such as DoS, Gray-hole, Sink-hole and Black-hole, the malicious node shows itself as a node with high resources (memory, energy, etc.) [12]. Therefore, it is important to consider energy to determine safe nodes. The difference between the remaining energy of a node and the node with the highest energy can be used to calculate the energy rate parameter. The low energy difference for a node indicates that it is likely to be malicious because it is assumed that malicious nodes declare their resources in high volumes. In order to calculate the energy rate more accurately, we consider it based on the initial energy. Accordingly, E_j is calculated by Eq. (7).

$$E_j = E_0 - [e_{max} - e_j] \quad (7)$$

where E_0 is the initial energy of the nodes, e_{max} is the node with the most energy and e_j is the energy of the j -th node.

Connectivity A fully-connected network can ensure safe data transfer [34]. Connectivity to a WSN requires that each node has at least one route available to connect to the sink. Basically, the connectivity is highly dependent on the location of the nodes. Here, the connectivity parameter is computed based on nodes with bi-directional links that can guarantee a fully-connected network. Accordingly, C_j is calculated by the Eq. (8).

$$C_j = \frac{nc_j}{L} \quad (8)$$

where nc_j refers to the number of links of the j -th node and L is the total number of connections in the network.

Trust score This parameter is defined based on the sum of direct trust and indirect trust, as shown in (9). T_j is the trust

score of the j -th node and is used to fill the ‘Trust’ field in the neighboring table.

$$T_j = \alpha \cdot DT_j + (1 - \alpha) \cdot IT_j \quad (9)$$

where DT_j and IT_j are direct trust and indirect trust related to the j -th node. Also, α is an impact coefficient for trust scores.

The direct trust score depends on the interactions between the two nodes. Therefore, each node in the network can estimate the trust of its neighbor nodes. Accordingly, DT_j is measured according to the Eq. (10).

$$DT_j = \beta \cdot \frac{ar_j}{nr_j} + (1 - \beta) \cdot \frac{at_j}{nt_j} \quad (10)$$

where ar_j is the number of acknowledgement packets received by the j -th node, and nr_j refers to the total number of packets received. Similarly, at_j and nt_j are related to packets sent from the j -th node. Moreover, β impact coefficient between packets sent and received to measure direct trust score.

Indirect trust depends on the behavior of the node in relation to its neighbors and is measured according to the data in the neighboring table. Accordingly, IT_j is defined according to the Eq. (11).

$$IT_j = \frac{\sum_{k \in nn_j} [T_k + T_k^j]}{|nn_j|} \quad (11)$$

where nn_j and $|nn_j|$ refers to set of neighboring nodes and their number, respectively. Also, T_k^j represents the recommended trust to s_j by s_k and T_k is the trust score of s_k .

4.3 Clustering and selection of CHs

Increasing network lifetime and improving energy consumption is a major challenge in WSN development [11, 12]. Clustering-based routing is recognized as an effective way to meet this challenge. Clustering in WSN involves grouping nodes into a number of clusters so that in each cluster one node plays the role of CH. The task of CHs is to collect and aggregate data from member nodes and then create a packet and transfer it to the sink. The appearance of clustering and CHs selection can help reduce energy consumption in routing and thus increase the lifetime of the network.

Applying node clustering at each routing round can increase the number of control packets, increase energy consumption, and increase network latency. For this reason, clustering is updated by the sink at specific timeslots (for example, each θ_{CA} routing round). Hitherto, various algorithms for clustering have been developed in WSN, among which evolutionary algorithms are very popular [1, 5]. In this

paper, ODMA is used as a novel evolutionary algorithm for clustering. ODMA is a novel meta-heuristic algorithm that takes advantage of the combination of both categories [13]. In the proposed scheme, in addition to the formation of clusters, the optimal number of clusters as well as CHs is determined. Here, the sink performs clustering and selects the CHs, and then informs the nodes of their details.

In ODMA, each solution is known as a software, and the optimization work is done by evolving the software. Software (leading or promising) evolves over time, and some become obsolete. In the open-source world, a promising software utilizes the efficient approaches of leading software to gain a better position in society. In general, the main operations of this algorithm include (1) moving towards leading software, (2) evolving leading software based on its history, and (3) branching out from leading software [13].

Each software in this problem is an appearance of node clustering. The structure of the software representation is a vector of real numbers of length N . Here, the index of each element refers to the corresponding node, and the content of each element determines the cluster number of the node. In order to accelerate ODMA convergence, the cluster number of each node is limited to 2 to \sqrt{N} [10]. In this regard, according to the defined encoding, the initial population is created randomly with N_p software.

The fitness of each software is calculated based on a multi-objective function. These objectives include (1) reducing the number of clusters, (2) increasing the intra-cluster density, (3) balancing the energy of the clusters, (4) balancing the candidate nodes in the clusters, and (5) reducing the distance of candidate nodes to the sink. Candidate nodes refer to safety nodes that are on the radio range of all members of their cluster. In fact, data exchange between candidate nodes and other members of the cluster can be done in a single-hop. Therefore, candidate nodes can be selected as CHs. In this paper, a candidate node with the highest level of safety is selected as CH from each cluster, where CHs are updated at each θ_{CH} routing round. The fitness function is calculated based on the objectives defined in Eq. (12).

$$\min w_1 \cdot K + w_2 \cdot D_v + w_3 \cdot \sigma_e + w_4 \cdot \sigma_c + w_5 \cdot D_C \quad (12)$$

where K is the total number of clusters in the i -th software. D_v is the sum of the intra-cluster distance, which is considered as the average for all clusters. σ_e is the standard deviation of the energy of the clusters and minimizing it causes a better energy balance between the clusters. σ_c is the standard deviation of the number of candidate nodes from all clusters and minimizing it helps to optimally select CHs. Finally, D_C is the average distance of all candidate nodes to the sink. Given the differences between the intended objectives, we use the total weight technique to apply the effect of each object. Here,

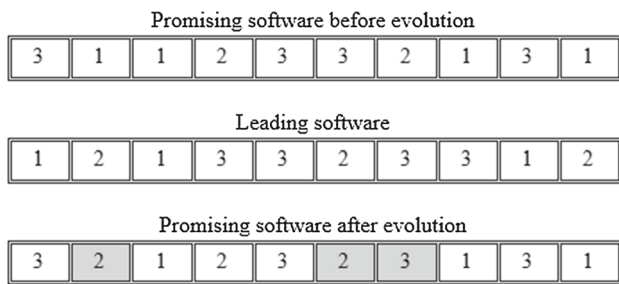


Fig. 7 Evolution of promising software based on leading software

w refers to the weight coefficient of each object in the fitness function, where $w_1 + w_2 + w_3 + w_4 + w_5 = 1$.

According to ODMA, software is sorted based on the value of the fitness function. Then, z softwares are selected as the leading softwares, while other softwares are promising. In general, the evolution of ODMA takes place in three stages. In the first stage, promising software is developed based on leading software. To do this, for each promising software, a leading software is randomly selected based on the fitness function and the evolution process is performed. Here, the concept of evolution is expressed by defining a possible variable, where each element of the promising software varies with the probability ρ according to its corresponding element in the leading software. In this regard, $\rho = 1$ makes the promising software completely similar to the leading software and $\rho = 0$ does not make any changes in the promising software. Figure 7 shows an example of this process based on $N = 10$ and $K = 3$, in which the clusters of nodes 2, 6 and 7 are changed based on the probability ρ .

In the second stage, the leading software evolves based on its history. Here, evolution is based on the current position (s_{cur}) and the latest position (s_{old}) of the leading software. Here, s_{new} is the new position of the leading software and is defined as Eq. (13).

$$s_{new,j} = \left\| \frac{(s_{cur,j} + s_{old,j})}{2} + Rand(-1, +1) \right\|, \quad \forall j = 1, 2, \dots, N \quad (13)$$

where $Rand$ is a random number generation function and $\| * \|$ is a rounding function. Also, j refers to the index of a node in software.

In the third stage, new software is produced from the leading software. Here, a number of weak software with minimal progress are removed and replaced by new software. The process of producing a new software from a leading software (s_r) is in accordance with Eq. (14).

$$s_{new,j} = \|s_{r,j} + Rand(-1, +1)\|, \quad \forall j = 1, 2, \dots, N \quad (14)$$

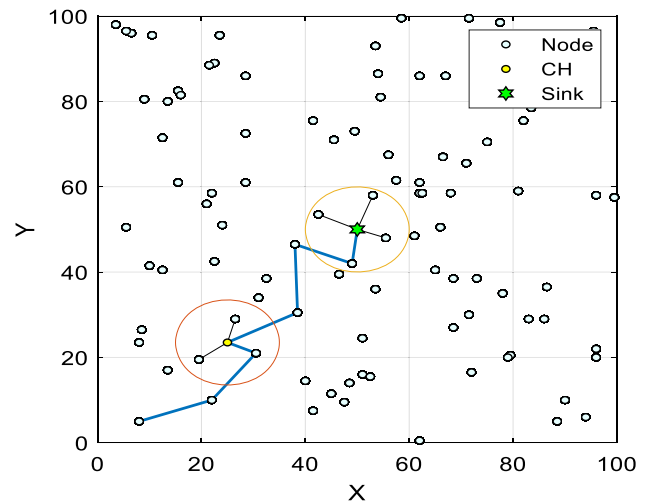


Fig. 8 Proposed routing process

After completing the clustering process and determining the CHs, the sink informs the role of the nodes by sending Hello packets. Therefore, each node is aware of its role and also knows its CH.

4.4 Proposed routing algorithm

In this section, the details of the proposed routing algorithm are described. The proposed routing is based on clustering, i.e., routing is done through CHs. Hence, the environment data is first sensed by the nodes of a cluster and then this data is sent to CH. All member nodes on their CH radio range can send data in a single-hop. However, nodes outside the CH radio range must select another suitable node based on the routing table to send data to CH. After receiving the data, CH aggregates and compresses the data and build a data packet based on them. Thereupon, CH sends the data packet to the sink. When CH is on the sink radio range, this transmission is done as a single-hop, otherwise a multi-hop route is specified for transmission. An overview of the proposed routing process is shown in Fig. 8.

Routing in EATMR is based on AODV protocol and multi-path routes approach. The main purpose of routing is to create a safe route for transferring data from CH to the sink in multi-hop mode. In AODV protocol based on multi-path routes, route request packet is sent from multiple routes to the destination, which leads to the discovery of different routes. Therefore, in the AODV protocol, RREQ is sent only to safe neighbor nodes instead of to all neighbors.

In addition, the EATMR uses a multi-path routing technique. According to the AODV protocol, RREQ can be

sent to the destination from several different routes, therefore, the source receives several RREP packets so that each of them can be one route. In the original AODV, the shortest route for routing is always specified. However, the EATMR analyzes all routes and determines the optimal and safe route based on the parameters of energy, trust, hop-count and distance. Based on these parameters, EATMR seeks to increase reliability and energy savings by choosing a shorter route. Therefore, EATMR is an energy-aware trust algorithm that uses only safe nodes in routing. The fitness function is formulated to select the route by Eq. (15). This function is calculated for each route and the route with the maximum value is considered for routing.

$$\max \xi_1.E_r + \xi_2.T_r - \xi_3.HC_r - \xi_4.D_r \quad (15)$$

where E_r , T_r , and D_r are the sum of the energies, trust scores, and distances of the nodes participating in route r , respectively, and HC_r refers to the hop-count in this route. Here, ξ is the weighting coefficients that determine the effect of each parameter, where $\xi_1 + \xi_2 + \xi_3 + \xi_4 = 1$.

4.5 EATMR algorithm

According to the above discussions, the proposed scheme is presented in the Algorithm 1.

Algorithm 1. Energy-Aware Trust based the Multi-path Routing (EATMR)

1. Start
 2. Network setup with N node (i.e., normal node, malicious node, and sink node) in a $m \times m$ meter environment
 3. Calculating the safety level of nodes based on parameters of energy, connectivity, and trust
 4. Identifying safe nodes based on safety level and threshold θ_{MN}
 5. Clustering the nodes based on ODMA and selecting CHs based on the trust score of candidate nodes
 6. Determining the safe node based on TDMA
 7. Collecting data by CH and creating data packet
 8. Discovering safe route by applying AODV protocol based multi-path routing
 9. Calculating the fitness for routes discovered with a hybrid function
 10. Selecting optimal route based on fitness values for transferring data packet to sink
 11. Transferring packet from source to destination based on selected route and updating nodes energy
 12. Updating safe nodes and safety levels based on threshold θ_{SN}
 13. Clustering and updating CHs based on thresholds θ_{CA} and θ_{CH}
 14. Repeating steps 6 to 13 to the end of the routing round
 15. End
-

5 Simulation results

In this section, the EATMR scheme is evaluated by performing simulations against schemes such as M-CSO [28] and SQUEER [20]. We perform extensive experiments and comparisons to demonstrate the effectiveness of EATMR in improving WSNs security. All experiments are performed on the Asus N551JK Notebook with specifications of Intel Core i7 processor at 3.5 GHz, 16 GB of RAM and Windows 10 operating system. The simulation was performed with MATLAB R2019a and the results were reported based on an average of 15 random deployments in WSN.

Here, popular metrics in WSN such as network lifetime, energy consumption, packet delivery rate, detection rate of malicious nodes, and number of alive nodes are used for comparison. The network lifetime is calculated by the number of packets sent before the death of the first node. Energy consumption refers to the total energy consumed, which in this article is reported as a percentage of total energy consumed relative to the total initial energy of the network. Packet delivery rate shows the number of successful packets sent per total number of packets sent. A packet is sent successfully if the acknowledgment message is received for it. Detection rate of malware nodes is defined as the number of malicious nodes detected relative to the total number of malicious nodes. Alive nodes refer to the number of nodes that are active in the network (nodes with energy). All of these metrics can be measured as long as the last node in the network is alive.

The continuation of this section consists of five subsections: simulation setup is discussed in the first subsection. The second subsection is related to EATMR analysis. The results of the comparisons are presented in the third subsection and discusses it in the fourth subsection.

5.1 Simulation setup

All nodes are homogeneous and have a fixed position, which are initially randomly deployed in an area of $100 \times 100m^2$. The sink node is always placed in the center of the area. Routing is based on the TDMA schedule, where in each round a node is randomly selected as the source node to sense the environment data and transfer it to the sink [10]. Let the source node not be a set of malicious nodes. The simulation is performed with a number of different nodes N (i.e., 25, 50, 100 and 200) in the presence of a DoS attack. The simulation is performed for 5000 routing rounds and packets are transmitted based on Constant Bit Rate (CBR) traffic type [35].

The initial trust score of all nodes is set to 1. In each scenario, R_{MN} is the rate of malicious nodes relative to the total number of nodes, which are evaluated with different values of 0.05, 0.1 and 0.2. Malicious nodes have abnormal

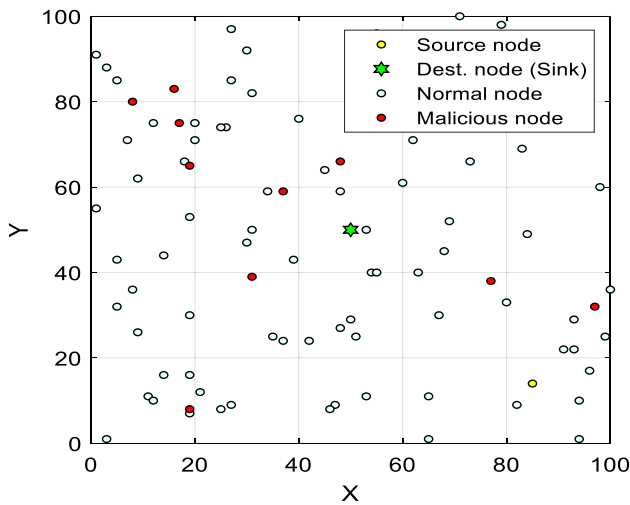


Fig. 9 Example of network topology

behaviors such as transmitting incorrect information, changing packet size, sending packets consecutively, and prevent sending packet. An example of the network topology tested with 90 normal nodes and 10 malicious nodes is shown in Fig. 9.

The efficiency and effectiveness of meta-heuristic algorithms depend on the precise setting of the parameters. In most research studies, the values of the parameters are adjusted based on the literature references or trial and error. In this paper, we also often determine the values of the parameters based on the literature references. However, some parameters are determined from the proposed scheme using the Taguchi technique [36] to achieve the best solution. This method ensures the identification of effective parameters and levels with fewer experiments by providing balance among the orthogonal index, parameters, and levels. The purpose of Taguchi technique is to maximize the S/N ratio (signal-to-noise). Here, the values obtained for the parameters are calculated based on the standard table of orthogonal arrays L_{27} [38]. In this paper, the values assigned to the EATMR parameters based on [10, 19] as well as the Taguchi technique are as follows:

$$\alpha = 0.6, \beta = 0.5, \theta_{CA} = 50, \theta_{CH} = 25, \theta_{SN} = 10, \\ N_P = 15, z = 5, \rho = 0.2, w_1 = w_5 = 0.1, w_2 = 0.3, \\ w_3 = w_4 = 0.25, \xi_1 = \xi_2 = 0.3, \xi_3 = \xi_4 = 0.2$$

5.1.1 EATMR analysis

Clustering in EATMR is done by ODMA where the reason for its choice is superiority over some similar algorithms. Here, we show that ODMA performs better in node clustering compared to the Imperialist Competitive Algorithm

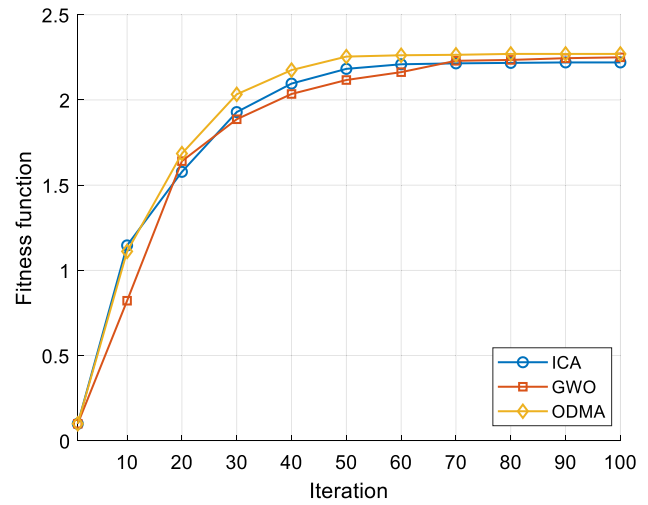


Fig. 10 Comparison of ODMA, ICA and GWO for node clustering

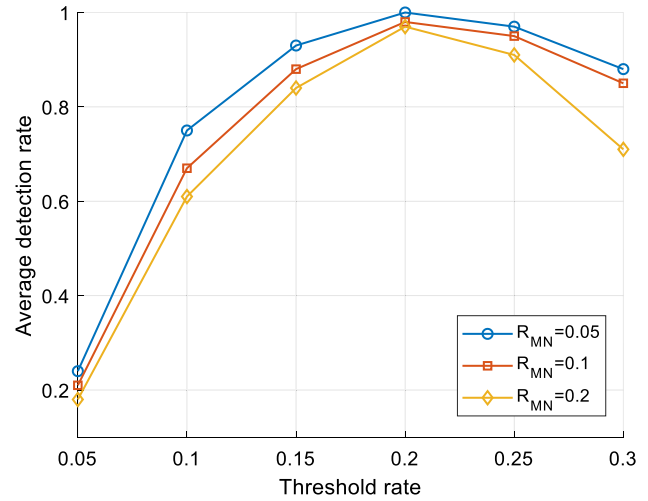


Fig. 11 detection rate of malicious nodes

(ICA) and the Gray Wolf Optimization (GWO). A comparison based on the fitness function defined in Eq. (12) for 100 iterations is presented in Fig. 10. The results of this comparison clearly show that ODMA has better convergence to achieve the optimal value of the objective function than other algorithms.

The EATMR scheme detects malicious nodes by computing the trust score and taking into account the θ_{MN} threshold. Here, the accuracy of detecting malicious nodes is analyzed based on the different threshold levels relative to the rate of different malicious nodes (i.e., R_{MN}). After determining the appropriate threshold, we justify the EATMR parameters accordingly. The accuracy of detecting malicious nodes based on different rates of θ_{MN} (i.e., 0.05–0.3) and R_{MN} (i.e., 0.05, 0.1 and 0.2) is shown in Fig. 11. The results of this simulation are presented with 100 nodes after 5000 routing rounds. As illustrated, most malicious nodes are identified

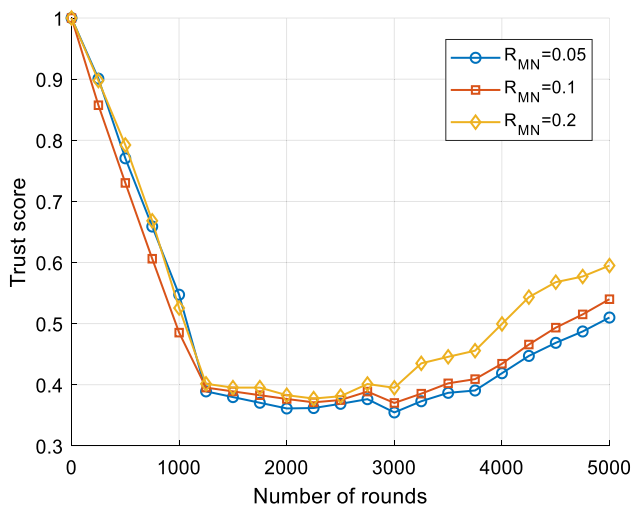


Fig. 12 Trust score of nodes relative to routing rounds

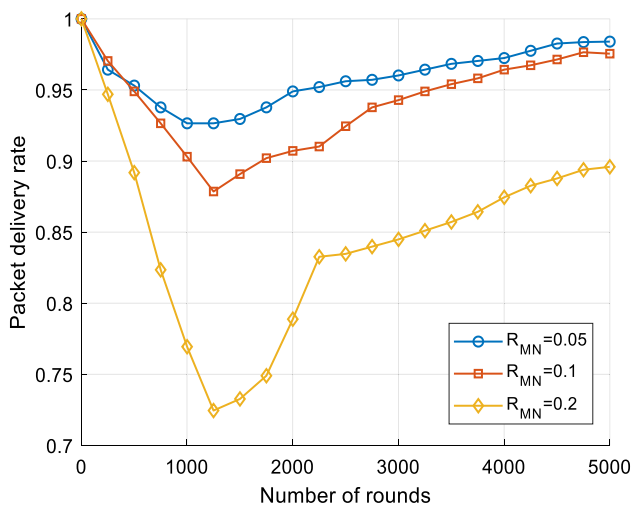


Fig. 13 Packet delivery rate relative to routing rounds

by $\theta_{MN} = 0.2$. In fact, thresholds smaller than 0.2 prevent the detection of malicious nodes, and thresholds greater than 0.2 identify normal nodes as malicious nodes.

The initial trust score of all nodes is 1, so the malicious nodes are not removed at first. As malicious nodes exhibit abnormal behaviors, the trust score as well as the packet delivery rate decrease. Therefore, during routing rounds and over time, malicious nodes are detected and slowly removed from the routing. As a result, abnormal behaviors in the network decrease, and trust scores and packets delivery rate increase. The results of the trust score and packet delivery rate relative to routing rounds in Figs. 12 and 13, respectively, confirm this. These results are presented for three different levels of malicious nodes (i.e., 0.05, 0.1 and 0.2), where the total number of nodes is 100. According to the results, it can be inferred that the network is more vulnerable in the

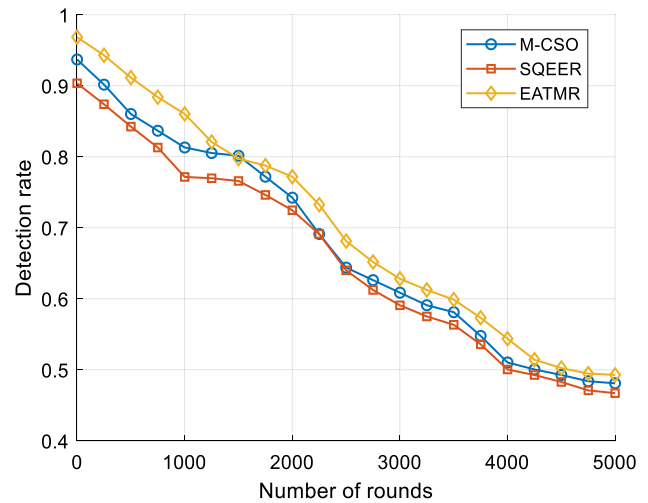


Fig. 14 Comparison of detection rate of malicious nodes relative to routing rounds

initial rounds, but with increasing rounds, this vulnerability decreases. In addition, increasing the rate of malicious nodes speeds up network vulnerabilities in the early rounds.

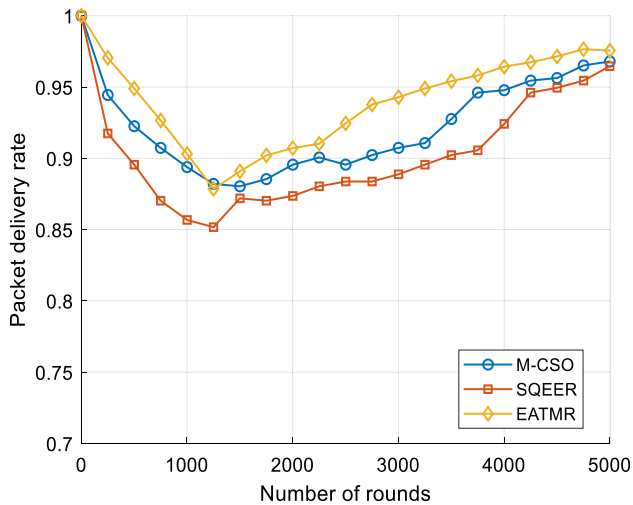
In order to provide a comprehensive analysis, the trust score and packet delivery rate for the number of different nodes (i.e., 20, 50, 100 and 200) with different levels of malicious nodes are reported at the end of routing rounds. This comparison is presented in Table 4 in the presence of DoS attacks. EATMR provides similar results for simulations with different number of nodes, however, as the number of nodes increases, the values of these metrics increased relatively. The reason for this could be the increase in the number of neighbors and thus access to more information about the nodes interactions to measure the trust score.

5.2 Comparison results

This section presents the results of various evaluation metrics to compare the M-CSO, SQEER and EATMR schemes. These factors indicate that the two algorithms are essentially similar to the proposed method and that is why we use them for comparison work. The comparisons of this section are based on $N = 100$, $\theta_{MN} = 0.2$ and $R_{MN} = 0.1$. The first comparison based on the detection rate of malicious nodes is shown in Fig. 14. At all rounds, the EATMR effectively detects malicious nodes and provides better security for the WSN compared to other schemes. As illustrated, the detection rates at the end of routing rounds for M-CSO, SQEER and EATMR are 0.481, 0.467 and 0.493, respectively. These results clearly show the superiority of the proposed scheme in detecting malicious nodes. The reason for this superiority is to identify malicious nodes based on the proposed trust score and prevent their presence in the routing process. However, as the number of rounds increases, the detection rate of

Table 4 EATMR evaluation based on trust score and packet delivery rate

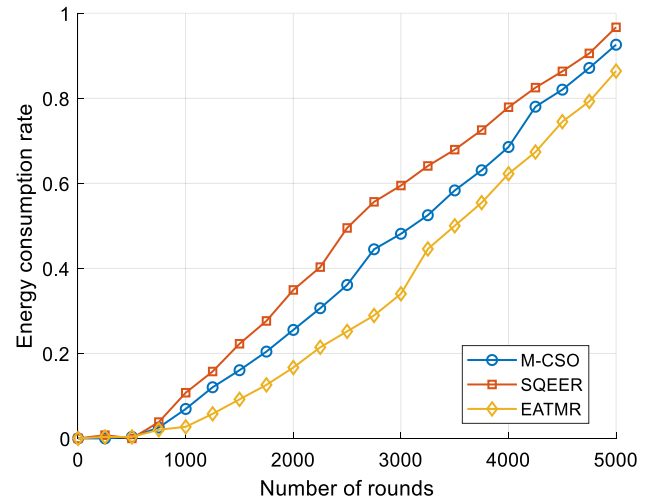
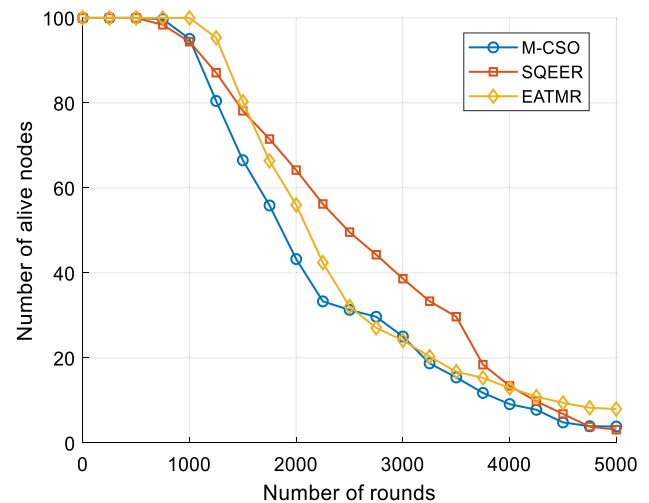
Number of nodes	Trust score			Packet delivery rate		
	0.05	0.1	0.2	0.05	0.1	0.2
25	0.416	0.46	0.503	0.893	0.875	0.801
50	0.443	0.479	0.533	0.92	0.907	0.826
100	0.508	0.542	0.595	0.984	0.975	0.896
200	0.526	0.55	0.607	0.997	0.979	0.887
Average	0.473	0.508	0.559	0.948	0.934	0.852

**Fig. 15** Comparison of packet delivery rate relative to routing rounds

malicious nodes decreases, which is due to the energy consumption and death of the sensor nodes.

A comparison of packet delivery rate is shown in Fig. 15 for different schemes in each routing round. As depicted, the results show the superiority of EATMR over M-CSO and SQEER. The reason for this superiority is the use of a trust-aware mechanism in clustering-based routing, which has identified malicious nodes and transmitted more packets. Evaluation of different schemes shows the superiority of EATMR with a packet delivery rate of 0.976. These results are 0.968 and 0.964 for M-CSO and SQEER, respectively.

Figure 16 shows the total energy consumed for different schemes in each routing round. Due to some EATMR features such as load balancing and reduced number of searches to discover the optimal route, the results are improved compared to other schemes. As illustrated, EATMR stores more energy than M-CSO and SQEER. Also, the EATMR tends to consume the same amount of energy per round of routing. At the end of the routing rounds, EATMR with 13.6% residual energy performs better than other schemes. Meanwhile, energy consumption compared to M-CSO and SQEER improved by 9.2 and 15.1%, respectively.

**Fig. 16** Comparison of network energy consumption relative to routing rounds**Fig. 17** Comparison of the number of alive nodes relative to routing rounds

In another experiment, EATMR as well as M-CSO and SQEER schemes were compared based on the number of alive nodes. The results of this comparison for the 5000 routing rounds are shown in Fig. 17. EATMR leads to a rapid

and simultaneous reduction of energy for most nodes, as this scheme provides a suitable distribution of energy consumption through clustering-based routing. Meanwhile, this reduction occurs later than other schemes. As illustrated, at the end of the routing, EATMR with 7 alive nodes is better than M-CSO and SQEER with 4 and 3 alive nodes, respectively.

According to the definition of network lifetime, here the first dead node is created for EATMR in the 1172 routing round. In this regard, the network lifetime for M-CSO and SQEER is 868 and 743, respectively. Therefore, EATMR achieves better network lifetime with better clustering, better CHs selection, and load balancing on nodes. The reason for this superiority is the use of trust-based routing, which detects malicious nodes and thus transmits more packets. Therefore, based on the increase in packet delivery rate and improved detection of malicious nodes, the energy consumption of nodes is reduced and this has led to an increase in network lifetime.

In order to provide a comprehensive comparison, the results of different schemes based on various metrics after 5000 routing rounds are shown in Table 5. This comparison is for M-CSO, SQEER and EATMR and is based on the number of different nodes (i.e., 20, 50, 100 and 200). In addition, we provide the average results for the number of different nodes in the last column to clarify the schemes performance. The results in most metrics with different number of nodes prove the superiority of EATMR. Therefore, EATMR is an effective scheme to increase WSN security.

5.3 Discussion

This section discusses the evaluation and comparison of the proposed scheme with the most relevant prior works. Comparisons and analyzes have been performed with different scenarios of the number of network nodes and the level of malicious nodes in the presence of DoS attack. EATMR results have been compared with M-CSO [27] and SQEER schemes based on various metrics such as network lifetime, energy consumption, packet delivery rate, malicious nodes detection, and number of alive nodes.

Compared to M-CSO and SQEER, EATMR improves network lifetime by taking the residual energy and safe level of nodes into account in selecting the potential CH candidates. Here, nodes with more trust and energy, less distance to the sink, and less distance to other cluster members can be more likely to be selected as CH. In general, improving network lifetime is because of the strength of EATMR in maintaining an energy balance in the WSN. As shown in Fig. 17, the EATMR has the best performance over the network lifetime with 1172 packets sent. Based on these results, EATMR enhances the network lifetime compared to M-CSO by 35% and SQEER by 58%. This superiority is a confirmation of the

power of cooperation between different nodes in estimating the trust score and identifying malicious nodes.

Meanwhile, most nodes can work together for longer rounds, and then almost all the energies are drained together and die. This is clearly observable in Fig. 17, where the number of alive nodes drops sharply, and the nodes tend to die in groups instead of dying separately. This is while the line for M-CSO and SQEER is gradually decreasing. Therefore, most nodes in these schemes die in the early rounds, and the EATMR fixes this defect. After 2000 rounds of routing since the death of the first node in EATMR, more than 85% of the nodes are discharged and removed. In addition, analyzes show that when the rate of malicious nodes increases, the number of alive nodes decreases, which is due to a decrease in the number of reliable nodes in the routing. Increasing the rate of malicious nodes leads to sending packets with more hop-counts, which consumes more energy and thus reduces the number of alive nodes.

M-CSO analysis showed that it suffers from slow convergence because its structure is based on MBO and CSO algorithms with a complex combination mechanism. This computational complexity has reduced packet delivery rate and thus reduced network lifetime. In this regard, SQEER evaluation indicates that it sends successive Hello packets to detect routing. This leads to the depletion of energy of many nodes, and, as a result, the energy consumption of the network increases. In addition, the routing process in M-CSO and SQEER is single-route, and there are no alternative routes. Hence, most of the routes discovered in these schemes have longer distances, and this is a reason for the increase in collision and failure due to the presence of malicious nodes. In contrast, EATMR simultaneously seeks multiple routes; so, if a route is not available, another route would be replaced. The comparison in Fig. 15 clearly shows that the proposed scheme has a higher packet delivery rate than M-CSO and SQEER. Also, the analysis shows that as the rate of malicious nodes increases, the packet delivery rate for all schemes decreases, which is due to the reduction in the number of reliable nodes in the routing.

The reason for the superiority of EATMR in detecting malicious nodes is the use of different factors in computing the distributed trust score. In addition to the trust score, the EATMR uses information about nodes' behavior with their neighbors to compute the level of node safety. Using energy to measure trust scores has made EATMR an intelligent energy-aware trust management scheme. The results of Fig. 14 show that the proposed scheme can ensure WSN security by effectively detecting malicious nodes and not participating them in routing. Malicious nodes in EATMR are identified on the basis of a threshold, where according to the developed AODV protocol, these nodes cannot participate in routing. In general, M-CSO performs better than SQEER, but poorer than EATMR due to the lack of applying alternative

Table 5 Comparison of M-CSO, SQEER and EATMR based on different metrics

Metrics	Schemes	Number of nodes				
		25	50	100	200	Average
Network lifetime	M-CSO	460	774	868	1706	952
	SQEER	384	591	743	1532	812.5
	EATMR	518	826	1172	1804	1080
Energy consumption rate	M-CSO	1	1	0.926	0.568	0.8735
	SQEER	1	1	0.967	0.606	0.8933
	EATMR	1	0.998	0.864	0.514	0.844
Packet delivery rate	M-CSO	0.869	0.905	0.968	0.984	0.9315
	SQEER	0.861	0.89	0.964	0.977	0.923
	EATMR	0.875	0.907	0.976	0.979	0.9343
Detection rate of malicious nodes	M-CSO	0.41	0.417	0.481	0.74	0.512
	SQEER	0.394	0.406	0.467	0.711	0.4945
	EATMR	0.415	0.433	0.493	0.763	0.526
Number of alive nodes	M-CSO	0	0	4	90	23.5
	SQEER	0	0	3	81	21
	EATMR	0	1	7	97	26.25

routes. The superiority of EATMR can be seen in the use of the multi-path routes technique, the safe selection of CHs, and the intelligent computation of the trust score.

6 Conclusion and future work

Considering security in WSN is a challenging task due to the presence of malicious nodes. Due to the limitations of WSNs, cryptographic techniques for security are highly complex and would not provide the expected performance. However, trust-aware routing schemes can provide better security with less complexity. In this regard, trust management models analyze sensor nodes for reliable routing and prevention of adverse effects against malicious nodes.

In this paper, the EATMR scheme for safe routing in WSN is introduced. EATMR is a multi-path routing algorithm based on the AODV protocol that uses an energy-aware trust model to discover the optimal route. The AODV protocol determines the optimal and safe route based on various parameters such as energy, trust, hop-count, and distance. Furthermore, EATMR uses clustering-based routing techniques to improve energy consumption and enhance network lifetime. Here, clustering is performed by ODMA and a multi-objective function to select CHs. The performance of EATMR has been assessed through simulations in the presence of a DoS attack. The results show that the proposed scheme improves the primary metrics such as energy consumption, packet delivery rate, and network lifetime compared to similar algorithms. Accordingly, EATMR shows

an average of 4.3 and 6.1% superiority over M-CSO and SQEER in different scenarios, respectively. For future work, EATMR can be evaluated on mobile multi-sink WSNs with energy limitation. Here, sinks can approach a set of low-energy nodes.

Funding The authors have not disclosed any funding.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Rezaeipanah, A., Nazari, H., & Ahmadi, G. (2019). A hybrid approach for prolonging lifetime of wireless sensor networks using genetic algorithm and online clustering. *Journal of Computing Science and Engineering*, 13(4), 163–174.
2. Sumathi, K., & Pandiaraja, P. (2020). Dynamic alternate buffer switching and congestion control in wireless multimedia sensor networks. *Peer-to-Peer Networking and Applications*, 13(6), 2001–2010.
3. Rostami, M., Berahmand, K., Nasiri, E., & Forouzandeh, S. (2021). Review of swarm intelligence-based feature selection methods. *Engineering Applications of Artificial Intelligence*, 100, 104210.
4. Ghobaei-Arani, M., & Shahidinejad, A. (2021). An efficient resource provisioning approach for analyzing cloud workloads: A metaheuristic-based clustering approach. *The Journal of Supercomputing*, 77(1), 711–750.

5. Berahmand, K., Nasiri, E., Rostami, M., & Forouzandeh, S. (2021). A modified DeepWalk method for link prediction in attributed social network. *Computing*, 103(10), 2227–2249.
6. Ghobaei-Arani, M. (2021). A workload clustering-based resource provisioning mechanism using Biogeography based optimization technique in the cloud-based systems. *Soft Computing*, 25(5), 3813–3830.
7. Rajpoot, P., & Dwivedi, P. (2020). Optimized and load balanced clustering for wireless sensor networks to increase the lifetime of WSN using MADM approaches. *Wireless Networks*, 26(1), 215–251.
8. Patel, T., & Kamboj, P. (2015). Opportunistic routing in wireless sensor networks: A review. In 2015 IEEE International Advance Computing Conference (IACC) (pp. 983–987). IEEE.
9. Shakarami, A., Ghobaei-Arani, M., Masdari, M., & Hosseinzadeh, M. (2020). A survey on the computation offloading approaches in mobile edge/cloud computing environment: A stochastic-based perspective. *Journal of Grid Computing*, 18(4), 639–671.
10. Rezaeipannah, A., Amiri, P., Nazari, H., Mojarad, M., & Parvin, H. (2021). An Energy-Aware Hybrid Approach for Wireless Sensor Networks Using Re-clustering-Based Multi-hop Routing. *Wireless Personal Communications*, 1–22.
11. Hajjee, M., Fartash, M., & Eraghi, N. O. (2021). An Energy-Aware Trust and Opportunity Based Routing Algorithm in Wireless Sensor Networks Using Multipath Routes Technique. *Neural Processing Letters*, 1–24.
12. Zahedi, A., & Parma, F. (2019). An energy-aware trust-based routing algorithm using gravitational search approach in wireless sensor networks. *Peer-to-Peer Networking and Applications*, 12(1), 167–176.
13. Hajipour, H., Khormuji, H. B., & Rostami, H. (2016). ODMA: A novel swarm-evolutionary metaheuristic optimizer inspired by open-source development model and communities. *Soft Computing*, 20(2), 727–747.
14. Chakeres, I. D., & Belding-Royer, E. M. (2004). AODV routing protocol implementation design. In 24th International Conference on Distributed Computing Systems Workshops, 2004, March. Proceedings. (pp. 698–703). IEEE.
15. M Badr, M. M., Ibrahim, M. I., Mahmoud, M., Fouda, M. M., Alsolami, F., & Alasmay, W. (2021). Detection of False-Reading Attacks in Smart Grid Net-Metering System. *IEEE Internet of Things Journal*. In press.
16. Fang, W., Zhang, W., Chen, W., Pan, T., Ni, Y., & Yang, Y. (2020). Trust-based attack and defense in wireless sensor networks: a survey. *Wireless Communications and Mobile Computing*. <https://doi.org/10.1155/2020/2643546>
17. Gilbert, E. P. K., Kaliaperumal, B., Rajsingh, E. B., & Lydia, M. (2018). Trust based data prediction, aggregation and reconstruction using compressed sensing for clustered wireless sensor networks. *Computers & Electrical Engineering*, 72, 894–909.
18. Ghugar, U., Pradhan, J., Bhoi, S. K., & Sahoo, R. R. (2019). LB-IDS: Securing wireless sensor network using protocol layer trust-based intrusion detection system. *Journal of Computer Networks and Communications*. <https://doi.org/10.1155/2019/2054298>
19. Zhao, J., Huang, J., & Xiong, N. (2019). An effective exponential-based trust and reputation evaluation system in wireless sensor networks. *IEEE Access*, 7, 33859–33869.
20. Kalidoss, T., Rajasekaran, L., Kanagasabai, K., Sannasi, G., & Kannan, A. (2020). QoS aware trust based routing algorithm for wireless sensor networks. *Wireless Personal Communications*, 110(4), 1637–1658.
21. Wu, X., Huang, J., Ling, J., & Shu, L. (2019). BLTM: Beta and LQI based trust model for wireless sensor networks. *IEEE Access*, 7, 43679–43690.
22. Anwar, R. W., Zainal, A., Outay, F., Yasar, A., & Iqbal, S. (2019). BTEM: Belief based trust evaluation mechanism for Wireless Sensor Networks. *Future Generation Computer Systems*, 96, 605–616.
23. Nie, S. (2019). A novel trust model of dynamic optimization based on entropy method in wireless sensor networks. *Cluster Computing*, 22(5), 11153–11162.
24. Jinhui, X., Yang, T., Feiyue, Y., Leina, P., Juan, X., & Yao, H. (2018). Intrusion detection system for hybrid DoS attacks using energy trust in wireless sensor networks. *Procedia Computer Science*, 131, 1188–1195.
25. Firoozii, F., Zadorozhny, V. I., & Li, F. Y. (2018). Subjective logic-based in-network data processing for trust management in collocated and distributed wireless sensor networks. *IEEE Sensors Journal*, 18(15), 6446–6460.
26. Janani, V. S., & Manikandan, M. S. K. (2018). Efficient trust management with Bayesian-Evidence theorem to secure public key infrastructure-based mobile ad hoc networks. *EURASIP Journal on Wireless Communications and Networking*, 2018(1), 1–27.
27. Sahoo, R. R., Ray, S., Sarkar, S., & Bhoi, S. K. (2018). Guard against trust management vulnerabilities in wireless sensor network. *Arabian Journal for Science & Engineering (Springer Science & Business Media BV)*, 43(12), 7229–7251.
28. Patil, P. A., Deshpande, R. S., & Mane, P. B. (2020). Trust and opportunity based routing framework in wireless sensor network using hybrid optimization algorithm. *Wireless Personal Communications*, 115(1), 415–437.
29. Khan, T., Singh, K., Hasan, M. H., Ahmad, K., Reddy, G. T., Mohan, S., & Ahmadian, A. (2021). ETERS: A comprehensive energy aware trust-based efficient routing scheme for adversarial WSNs. *Future Generation Computer Systems*, 125, 921–943.
30. Sun, B., & Li, D. (2017). A comprehensive trust-aware routing protocol with multi-attributes for WSNs. *IEEE Access*, 6, 4725–4741.
31. Wang, R., Zhang, Z., Zhang, Z., & Jia, Z. (2018). ETMRM: An energy-efficient trust management and routing mechanism for SDWSNs. *Computer Networks*, 139, 119–135.
32. Mehete, D. C., Roslin, S. E., & Wagh, S. J. (2019). Detection and prevention of black hole and selective forwarding attack in clustered WSN with active trust. *Cluster Computing*, 22(1), 1313–1328.
33. Anchugam, C. V., & Thangadurai, K. (2015). Detection of black hole attack in mobile ad-hoc networks using ant colony optimization-simulation analysis. *Indian Journal of Science and Technology*, 8(13), 1–10.
34. Boukerche, A., & Sun, P. (2018). Connectivity and coverage based protocols for wireless sensor networks. *Ad Hoc Networks*, 80, 54–69.
35. Roslin, S. E. (2021). Data validation and integrity verification for trust based data aggregation protocol in WSN. *Microprocessors and Microsystems*, 80, 103354.
36. Chen, C. S., Lin, J. M., Lee, C. T., & Lu, C. D. (2014). The hybrid Taguchi-Genetic algorithm for mobile location. *International Journal of Distributed Sensor Networks*, 10(3), 489563.

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