Soft Computing (2023) 27:11731–11754
https://doi.org/10.1007/s00500-023-07856-z

APPLICATION OF SOFT COMPUTING

Trust-aware and energy-efficient data gathering in wireless sensor networks using PSO

Keiwan Soltani[1] · Leili Farzinvash[1] · Mohammad Ali Balafar[1]

Accepted: 17 January 2023 / Published online: 7 February 2023

Abstract

Trustworthiness and energy efficiency are two important aspects of data gathering in Wireless Sensor Networks (WSNs). The first criterion can be fulfilled by adopting trustworthy nodes for data communication along with choosing watchdogs for monitoring. Additionally, the use of the clustering scheme reduces energy exhaustion substantially. Accordingly, effective data gathering requires trust-aware and energy-efficient clustering, data gathering tree construction, and watchdog selection. The previous data gathering algorithms did not include all of the clustering, tree construction, and watchdog selection phases. Furthermore, some studies proposed greedy schemes to solve the mentioned phases and had low performance. In this paper, we propose the Trust-aware and Energy-efficient Data Gathering (TEDG) algorithm to gather data more effectively. The proposed scheme comprises all the above-mentioned phases, including clustering, tree construction, and watchdog selection. These phases are modeled as optimization problems, and they are solved using Particle Swarm Optimization (PSO). The watchdog selection phase has variable-length particles because the number of watchdogs has not been unknown. Novel particle representation and initialization schemes are proposed to handle these particles. According to the performed simulations, TEDG improves consumed energy for data delivery to the sink, standard deviation of the residual energy of nodes, and network lifetime by 220%, 81%, and 129%, respectively.

Keywords Wireless sensor network · Trust · Energy efficiency · Clustering · Watchdog · Particle swarm optimization

1 Introduction

Nowadays, the use of Wireless Sensor Networks (WSNs) has increased (Singh et al. 2018). The nodes within a WSN monitor their surrounding area and send their sensed data in a multi-hop manner. Data transmission to the sink against different attacks, such as data modification, Black Hole Attack (BHA), and Selective Forward Attack (SFA), must be protected to ensure effective monitoring of the environment (Tomić and Mccann 2017; Ansari et al. 2021). BHA and SFA, in which the malicious nodes drop data packets, are among the most prominent attacks and should be mitigated to ensure effective data gathering. A practical approach for this purpose is to use more trusted nodes for data transmission (Cai et al. 2019; Khalid et al. 2019; Han et al. 2022). In addition, the use of watchdogs provides effective monitoring throughout the network. These nodes monitor the activity of others within the network (Monnet et al. 2017). The malicious nodes can be determined using the information gathered by watchdogs. The amount of sent data to the sink can be increased by preventing the malicious nodes from participating in the data gathering process (Bangotra et al. 2021; Shahid et al. 2022).

Another important criterion in designing WSNs is energy efficiency. The energy of sensors is limited and will be exhausted after a while. Therefore, measures should be taken to ensure that the nodes have the least energy consumption rate. Exploiting the clustering technique, which has been applied in many studies such as (Ni et al. 2017; Mittal et al. 2021), reduces energy consumption

✉ Leili Farzinvash
l.farzinvash@tabrizu.ac.ir

Keiwan Soltani
keiwansoltani95@ms.tabrizu.ac.ir

Mohammad Ali Balafar
balafarila@tabrizu.ac.ir

[1] Present Address: Faculty of Electrical and Computer Engineering, University of Tabriz, Bahman Blvd, PO Box: 516661647, Tabriz, Iran

considerably. In this scheme, the Cluster Head (CH) per cluster aggregates the gathered data by the cluster members. Consequently, the amount of data and required energy for its transmission decreases substantially. The gathered data by CHs should be forwarded to the sink. For this aim, a data gathering tree is constructed over the CHs and some relay nodes (Elhabyan and Yagoub 2015; Khalid et al. 2019; Pavani and Rao 2019). A data gathering algorithm should take into account the trust and energy criteria to decrease the packet loss and increase the network lifetime. It is essential to consider these measures in both the cluster and data gathering tree construction schemes. Moreover, some watchdogs are required to monitor the packet forwarding by sensor nodes and identify malicious ones. In summary, an effective data gathering algorithm comprises trust-aware and energy-efficient clustering, routing, and watchdog selection schemes.

As mentioned above, many trust and energy-aware data gathering algorithms have been proposed so far. However, these schemes did not include all of the above-mentioned phases. For example, references (Elhabyan and Yagoub 2015; Edla et al. 2019; Shyama et al. 2022) did not address the security challenges and are vulnerable to SFA. The proposed trust-aware algorithms in Yun et al. (2018), Shcherba et al. (2019) and Bangotra et al. (2021) did not apply the clustering scheme. Another concern is about the problem-solving approach of the proposed algorithms. Some studies used greedy approaches to solve the phases. The proposed trust-aware and cluster-based algorithms in Fang et al. (2021), Hu et al. (2021), Isaac Sajan and Jasper 2021 and Yang et al. 2021) were greedy and did not yield high-throughput solutions. On the other hand, the approaches which applied optimization algorithms did not pay enough attention to optimizing the clusters and data gathering tree in terms of trust and energy (Pavani and Rao 2019; Rodrigues and John 2020; Han et al. 2022; Supriya and Adilakshmi 2022). They did not concern with trustworthiness and energy efficiency in both clustering and routing schemes. Furthermore, they did not use watchdogs, and the trust level of each node was derived from its next hop on the tree.

Finally, the proposed approaches for watchdog selection had low performance. The proposed schemes in Mittal et al. (2021), Shahid et al. (2022) did not specify watchdogs and computed trust values of nodes based on the sensors in their neighborhood and clusters, respectively. This monitoring strategy requires a considerable amount of energy. Monnet et al. (2017) selected a small number of high-energy nodes as watchdogs to reduce the overhead caused by monitoring. This approach, however, did not necessarily assign watchdogs to important nodes such as CHs. References (Bouali et al. 2016; Abdellatif and Mosbah 2020; Hu et al. 2021) proposed effective and energy-efficient monitoring schemes by assigning watchdogs to more important nodes such as CHs and relay nodes. The drawback of these algorithms was that they were greedy and could not select watchdogs properly.

Considering the shortcomings of the existing approaches, we propose the Trust-aware and Energy-efficient Data Gathering (TEDG) algorithm. The proposed scheme consists of clustering, tree construction, and watchdog selection phases. The mentioned phases consider trust and energy criteria. Accordingly, the objective of the clustering phase is to adopt nodes with high trust value and residual energy as CHs. The main concern in the proposed tree construction scheme is to choose relay nodes from trusted and high-energy sensors. Finally, in the proposed watchdog selection scheme, sensors with high trust and energy levels are adopted to perform monitoring tasks. Each phase can be considered as an optimization problem, which aims to increase the trust and energy level of adopted nodes as much as possible. We use an extension of Particle Swarm Optimization (PSO), namely PSO-TVAC (Ratnaweera et al. 2004), to solve these optimization problems. Specifically, we design the PSO components per phase, including particle representation, fitness function, and decoding procedure. The particles of the clustering phase represent the selected CHs. Moreover, the particles of the tree construction phase denote the priority of sensors to be selected as relay nodes. Finally, an assumed particle in the watchdog selection phase represents the selected watchdogs for monitoring CHs and relay nodes. The CHs are responsible for monitoring the cluster members to reduce the overhead caused by monitoring. To sum up, the key contributions of the TEDG algorithm can be listed as follows:

1. The proposed algorithm includes trust-aware and energy-efficient clustering, data gathering tree, and watchdog selection phases. Therefore, it yields high-throughput solutions.
2. All phases are modeled and solved by PSO-TVAC. The proposed clustering and tree construction schemes remove the shortcomings of the existing approaches and include proper particle representation, fitness function, and decoding procedure. Additionally, to the best of our knowledge, meta-heuristic algorithms have not been applied for watchdog selection before.
3. As the number of watchdogs has not been known, the particles of the watchdog selection phase are of variable length. We propose novel initialization and particle updating procedures to handle the mentioned particles. Furthermore, some randomness is included in the watchdog selection scheme to reduce the possibility of selecting the malicious nodes as watchdogs.

4. Extensive simulations demonstrate the superiority of the TEDG algorithm over existing schemes.

The rest of the paper is organized as follows. The related studies to our research are discussed in Sect. 2. The system model is presented in Sect. 3. The PSO-TVAC algorithm is described in Sect. 4. Next, TEDG is explained in Sect. 5. The proposed algorithm is evaluated in Sect. 6. Finally, Sect. 7 concludes the paper.

2 Related works

The related research to our work can be categorized into trust-aware and energy-aware data gathering algorithms. Some studies only considered trust or energy criterion, while others investigated both criteria and proposed trust and energy-aware schemes.

A critical concern in design systems is to ensure their reliability and security (Gunjan et al. 2015; Swapnarani et al. 2022). This topic has been deeply studied in WSNs (Prabhu and Mary Anita 2020). Yun et al. (2018) modified the Dijkstra algorithm for trust-aware routing, where the cost of a link was defined based on the level of mistrust of the receiver node to the sender one. In the algorithm presented by Khalid et al. (2019), each node adopted the next hop toward the sink based on the trust criterion. The trust value and the amount of residual energy were applied to determine the next hop in Bangotra et al. (2021). The proposed algorithm by Wang et al. (2014) extended AODV. In this study, the cost of a link was defined based on the trust level of its end nodes and some QoS parameters such as reliability and delay. The AODV algorithm was also used in Yin et al. (2022). This study clustered the sensors first, followed by performing multi-path AODV-based routing to connect CHs to the sink. The considered criteria in this study were trustworthiness and energy efficiency. In Isaac Sajan and Jasper (2021), the considered measures for cluster construction and routing were residual energy of nodes, required energy for data transmission, trust value of nodes, and data transmission delay. The proposed algorithm identified the malicious nodes based on their trust value and completely isolated them from the network. Reference (Sánchez-Casado et al. 2015) noticed that packet loss may occur due to collision, mobility of nodes, and SFA. It estimated the probability of collision, mobility, and packet loss to derive the probability of SFA occurrence. A node was assumed to be malicious if the derived probability for SFA occurrence exceeds a pre-defined threshold.

Some proposals investigated the topic of trust in cluster-based WSNs. The algorithm proposed in Bouali et al. (2016) employed the clustering scheme for data communication. In this study, the CHs and border nodes were responsible for data routing among the clusters. Some watchdogs monitored the data routing procedure to ensure reliable data delivery. In addition, the CH and watchdog selection was performed considering the trust criterion. Reference (Saidi et al. 2020) clustered nodes to reduce energy exhaustion. The criteria for CH selection were residual energy, trust level, and the number of neighbors per node. In the proposed monitoring scheme, the CH and members of each cluster monitored each other. The disadvantage of this plan was the high energy depletion rate of the CHs due to direct data transmission to the sink. Monnet et al. (2017) applied LEACH (Heinzelman et al. 2002) to construct clusters. This study adopted some watchdogs based on the residual energy criterion. Additionally, some random guards were specified to monitor the watchdogs and prevent malicious ones from reporting false information.

Yang et al. (2021) extended LEACH (Heinzelman et al. 2002) for cluster construction. Furthermore, they used the cryptography technique to identify malicious nodes. For this aim, some encrypted information was added to the header of each forwarded packet. The sink recognized malicious nodes based on the received evidence. This scheme, however, yielded considerable overhead. The LEACH algorithm was also used for clustering in Li et al. (2019; Abdellatif and Mosbah 2020). These studies considered trust and energy criteria for CH and watchdog selection. The proposed algorithm in Hu et al. (2021) performed clustering and routing along with watchdog selection. In this study, the most trusted nodes were adopted as CHs. Each CH selected one of the closer CHs to the sink as the next hop based on the trust and energy measures. The same criteria were applied to choose some watchdogs in the clusters to monitor the activities of CHs and cluster members.

References (Pavani and Rao 2019; Rodrigues and John 2020) applied meta-heuristic algorithms to mitigate SFA and increase energy efficiency. The k-means and modified monkey optimization algorithms were applied to select CHs in Supriya and Adilakshmi (2022). The considered metrics for this purpose were the trust value of nodes, their energy level, their distance from the sink, and the degree of the nodes. This algorithm consumed a considerable amount of energy due to direct data transmission from the CHs to the sink. Han et al. (2022) included trust and energy criteria in the proposed CH selection scheme in LEACH. Furthermore, Genetic Algorithm (GA) was used to construct a spanning tree over the CHs. The proposed algorithm in Rodrigues and John (2020) selected high-energy CHs first. Next, it combined chicken swam optimization with the dragonfly algorithm to construct highly trusted paths for intra-cluster data communication. Deterministic finite automata and PSO were applied to construct energy-efficient clusters and the data gathering tree in Prithi and Sumathi (2020). This study also proposed a greedy scheme to detect malicious nodes. These schemes had a low performance because they did not include energy and trust criteria in both phases. Furthermore, they did not adopt watchdogs and the trust level of each node was reported by its next hop on the tree.

The proposed algorithm in Pavani and Rao (2019) employed PSO and firefly algorithm for cluster and tree construction, respectively. This algorithm concerned the trust criterion in the cluster construction phase. Furthermore, only CHs are included in the data gathering tree. In this plan, CHs consumed a considerable amount of energy due to data transmission over long distances. Mittal et al. (2021) applied cuckoo search and fuzzy logic for cluster construction. The cuckoo search algorithm was used for modeling the CH selection problem, while fuzzy logic was applied for evaluating the fitness value of individuals. In the proposed tree construction scheme, for each CH, the closest neighboring CH to the sink was adopted as its next hop. Additionally, each node was monitored by its neighbors, which yielded considerable overhead. Sajan et al. (2022) firstly identified the members per cluster, followed by specifying the CHs. Next, the data transmission tree was constructed over CHs using gray wolf optimization algorithm. The considered criteria in this study were trust, energy, distance, and delay. Shahid et al. (2022) proposed a cellular automata-based SFA detection and prevention scheme to improve the security level of LEACH. This research focused on the trustworthiness of nodes and energy efficiency, which are of high importance under SFA.

The rest of this section briefly reviews energy-aware data gathering algorithms that did not concern the trust criterion. GA was employed for cluster construction in Mittal et al. (2019). In the proposed scheme, each gen of a given chromosome represents the status of its corresponding node, i.e., to be a CH or a member node. The aim was to use high-energy nodes as CHs and decrease the energy required for data gathering. Mann and Singh (2019) improved the Artificial Bee Colony (ABC) algorithm by enhancing its initialization scheme and integrating it with differential evolution. The improved ABC was applied for CH selection. Furthermore, a greedy scheme was proposed to determine the members of each cluster. The fuzzy logic was applied for cluster construction in Hou et al. (2022). The considered criteria in this study were the energy level of nodes, their distance from the sink, and the number of neighbors per node.

Elhabyan and Yagoub (2015) performed clustering and tree construction using PSO. In the cluster construction phase, a particle represented the set of CHs, aiming to increase energy efficiency. The proposed tree construction scheme assigned a priority per node and used these values to construct high-quality trees. The criteria for tree construction were the energy level of CHs, the number of non-CH nodes on the tree, and the link quality. The proposed algorithm in Shyama et al. (2022) integrated GA and PSO, and used the resultant meta-heuristic algorithm for path construction. The main concern of the path construction scheme was fault tolerance, while the proposed cluster construction approach aimed to improve energy-efficiency and coverage. Ant colony optimization was applied for tree construction in Arora et al. (2020), where the criterion for next-hop selection was the residual energy of sensors. Reference (Pachlor and Shrimankar 2018) proposed a greedy approach to balance energy exhaustion throughout the WSN. In each round of executing the algorithm, the current CHs adopted the next ones considering the energy consumption measure. Additionally, high-loaded clusters were decomposed into two clusters for better load balancing.

The studied algorithms are compared in Table 1. The measures for comparison are the inclusion of clustering, tree construction, and watchdog selection phases, the application of meta-heuristic algorithms for each phase, and included criteria per phase. It is noticeable that some algorithms assumed end-to-end communication and constructed routes instead of data gathering trees. Additionally, in some cluster-based algorithms, the CHs directly send data to the sink. Some trust-aware algorithms, which considered trust criteria in the clustering and tree construction, did not specify watchdogs. In these schemes, the trust value per node was calculated using the reports of its neighbors or corresponding CH. Finally, as shown in Table 1, some schemes choose watchdogs randomly.

3 System model

The system model comprises the network, adversary, and energy models as described in the following.

3.1 Network model

The considered network consists of n nodes that are randomly distributed in a monitoring area of dimensions $l \times w$. The transmission range and initial energy of the nodes are the same and indicated by tr and e_{init}, respectively. The notation $\boldsymbol{N_i}$ denotes the set of neighbors of node s_i, and d_{ij} presents the distance between nodes s_i and s_j. Furthermore, the residual energy of node s_i is shown by e_i. Additionally, the trust level of node s_i is presented by t_i. The sensors are partitioned into $k \times n$ clusters to enhance

Table 1 Comparison of the considered algorithms (Incl. = Inclusion, Aprch. = Approach, Tree = Tree construction, R = Routing, W = Watchdog selection, D = IDS, G = Greedy, M = Meta-heuristic, F = Fuzzy logic, RND = Random, T = Trust, E = Energy)

Reference	Clustering			Tree or routing			Watchdog selection		
	Incl.	Aprch.	Criteria	Incl.	Aprch.	Criteria	Incl.	Aprch.	Criteria
Abdellatif and Mosbah (2020)	✔	G	T, E	T	G	T, E	W	G	T, E
Arora et al. (2020)	–	–	–	T	M	T	–	–	–
Bangotra et al. (2021)	–	–	–	R	G	T, E	–	–	–
Bouali et al. (2016)	✔	G	T	R	G	T	W	G	T
Cai et al. (2019)	✔	G	T	–	–	–	–	–	–
Edla et al. (2019)	✔	G	E	Tree	M	E	–	–	–
Elhabyan and Yagoub (2015)	✔	M	E	Tree	M	E	–	–	–
Fang et al. (2021)	✔	G	T, E	–	–	–	–	–	–
Han et al. (2022)	✔	G	T, E	Tree	M	T, E	–	–	–
Heinzelman et al. (2002)	✔	G	E	–	–	–	–	–	–
Hou et al. (2022)	✔	F	E	–	–	–	–	–	–
Hu et al. (2021)	✔	G	T	Tree	G	T, E	W	G	T, E
Khalid et al. (2019)	–	–	–	Tree	M	T, E	–	–	–
Li et al. (2019)	✔	G	T	–	–	–	–	–	–
Shyama et al. (2022)	✔	M	E	Tree	M	E	–	–	–
Mann and Singh (2019)	✔	M	E	–	–	–	–	–	–
Mittal et al. (2021)	✔	M	T, E	Tree	G	E	–	–	–
Mittal et al. (2019)	✔	M, F	E	–	–	–	–	–	–
Monnet et al. (2017)	✔	G	E	–	–	–	W	RND	–
Ni et al. (2017)	✔	M	E	–	–	–	–	–	–
Pachlor and Shrimankar (2018)	✔	G	T	–	–	–	–	–	–
Pavani and Rao (2019)	✔	M	T, E	Tree	M	E	–	–	–
Prithi and Sumathi (2020)	✔	G	E	Tree	M	E	D	G	–
Isaac Sajan and Jasper (2021)	✔	G	E	Tree	G	E, T	–	–	–
Rodrigues and John (2020)	✔	M	E	Tree	M	T	–	–	–
Saidi et al. (2020)	✔	G	T, E	–	–	–	–	–	–
Sajan et al. (2022)	✔	G	T, E	Tree	M	T, E	–	–	–
Sánchez-Casado et al. (2015)	–	–	–	R	G	T	–	–	–
Shahid et al. (2022)	✔	G	T, E	–	–	–	–	–	–
Shcherba et al. (2019)	–	–	–	R	G	T	–	–	–
Supriya and Adilakshmi (2022)	✔	M	T, E	–	–	–	–	–	–
Wang et al. (2014)	–	–	–	R	G	T	–	–	–
Yang et al. (2021)	✔	G	E	–	–	–	D	G	–
Yin et al. (2022)	✔	M	E	R	G	T, E	–	–	–
Yun et al. (2018)	–	–	–	R	G	T	–	–	–
TEDG	✔	M	T, E	Tree	M	T, E	W	M	T, E

the energy efficiency, where parameter k shows the percentage of nodes that are adopted as CHs. The CH of the j $^{\text{th}}$ cluster, denoted as C_j, is presented by ch_j.

3.2 Adversary model

The considered attack in this paper is SFA, in which each malicious node drops some of the received packets. It is assumed that each malicious sensor drops a received packet with a probability α or forwards it to the next hop with a probability $1 - \alpha$. Furthermore, we make no assumption about the malicious nodes. Each sensor may be compromised by the attacker and become a malicious node. It is assumed that γ percentage of the deployed sensors are malicious.

The introduced notations are summarized in Table 2.

3.3 Energy model

The proposed model by Heinzelman et al. (2002) is applied to compute the amount of energy required to send and receive a packet of length b. These values are derived using (1) and (2), respectively. In these equations, E_{TX} and E_{RX} represent the amount of energy required to send and receive the packet, respectively. According to (1), the amount of energy consumed to send the packet depends on d_{ij}. If d_{ij} is less than the threshold (i.e., d_0), the free space model is used. Otherwise, the multi-path fading channel model is applied to calculate E_{TX}. Parameters ε_{fs} and ε_{mp} denote the exhausted energy by the amplifier for free space and multi-path fading channel models, respectively. Furthermore, e_c indicates the consumed energy by the electronic circuit.

$$E_{\mathrm{TX}}(b, d_{ij}) = \begin{cases} be_c + b\varepsilon_{fs}d_{ij}^2 & d_{ij} < d_0 \\ be_c + b\varepsilon_{mp}d_{ij}^4 & d_{ij} \geq d_0 \end{cases} \tag{1}$$

$$E_{\mathrm{RX}}(b) = be_c \tag{2}$$

The values of the parameters related to the energy model are given in Table 3.

4 Overview of the PSO-TVAC algorithm

PSO is a meta-heuristic scheme inspired by the social behavior of birds for food finding. In this algorithm, some particles search the solution space to find high-quality solutions. Particle P_i consists of the position vector $\boldsymbol{X}_i^t = [x_{i1}^t, x_{i2}^t, \ldots, x_{i,dim}^t]$ and the velocity vector $\boldsymbol{V}_i^t = [v_{i1}^t, v_{i2}^t, \ldots, v_{i,dim}^t]$, where parameters dim and t present the space dimension and the iteration number. The position and velocity vectors of particles are initiated randomly and updated in each iteration to increase (decrease) their fitness (cost). Each particle P_i is updated based on the best position found by the particle itself ($\boldsymbol{pbest}_i^t$) and the best position found by all particles so far ($\boldsymbol{gbest}^t$) using (3) and (4). The pbests of particles and the gbest are updated per iteration. The algorithm is continued until some termination conditions are met.

Table 2 The list of the used notations

Notation	Definition
n	Number of nodes
l, w	Dimensions of the area
tr	Transmission range of nodes
e_{init}	Initial energy of nodes
$\boldsymbol{N_i}$	Set of neighbors of s_i
d_{ij}	Distance between nodes s_i and s_j
e_i	Residual energy of node s_i
t_i	Trust level of node s_i
k	Percentage of nodes that are selected as CH
C_j	The j th cluster
ch_j	CH of C_j
α	Packet dropping probability of malicious nodes
γ	Percentage of malicious nodes

Table 3 The values of the parameters of the energy model

Parameter	Value
e_c	50 nJ/bit
ε_{fs}	10 pJ/bit/m^2
ε_{mp}	0.0013 pJ/bit/m^4
d_0	$\sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}$

$$\boldsymbol{V}_i^{t+1} = w\boldsymbol{V}_i^t + \mathrm{c}_1\boldsymbol{r}_{1i}(\boldsymbol{pbest}_i^t - \boldsymbol{X}_i^t) + \mathrm{c}_2\boldsymbol{r}_{2i}\left(\boldsymbol{gbest}^t - \boldsymbol{X}_i^t\right) \tag{3}$$

$$\boldsymbol{X}_i^{t+1} = \boldsymbol{X}_i^t + \boldsymbol{V}_i^{t+1} \tag{4}$$

where parameter w is the inertia weight, and parameters c_1 and c_2 are the self-cognition and social-influence coefficients, respectively. Additionally, $\boldsymbol{r}_{1i}$ and $\boldsymbol{r}_{2i}$ are random dim-dimensional vectors whose elements lie within the range $[0-1]$.

The main shortcoming of PSO is the tendency of the particles to fly toward the gbest, which yields trapping into local optimums. Many PSO variants have been proposed to improve its performance (Wang et al. 2018). Among them, we adopt PSO-TVAC (Ratnaweera et al. 2004), which has been applied to solve different optimization problems in WSNs (Zhao et al. 2017; Fang and Feng 2018; Wu et al. 2019). This PSO variant decreases w and c_1 and increases c_2 over time as stated in (5)-(7).

$$w = w_{\max} - t\frac{w_{\max} - w_{\min}}{t_{\max}} \tag{5}$$

$$c_1 = \frac{t}{t_{\max}}(c_{1e} - c_{1s}) + c_{1s} \tag{6}$$

$$c_2 = \frac{t}{t_{\max}}(c_{2e} - c_{2s}) + c_{2s} \tag{7}$$

where $t_{\max}$ is the maximum number of iterations. The values of the parameters used in PSO-TVAC are given in Table 4. Furthermore, the flowchart of this algorithm is shown in Fig. 1.

5 The TEDG algorithm

The operation of the proposed scheme is as follows. Initially, the network undergoes the bootstrapping process. During this process, each sensor is assigned a unique ID, and its location is determined. Furthermore, the nodes discover their neighbors using hello packets. The sink collects the locations and neighbor lists of the sensors for further consideration. Next, the network operation time is divided into rounds. Each round comprises network configuration, data gathering, and trust evaluation procedures. To configure the WSN, the sink executes the proposed clustering, tree construction, and watchdog selection schemes. After that, it notifies the sensors about their roles –CH, relay node, watchdog, or ordinary node—in the current round. Finally, the sink evaluates the trust values of sensors using the sent reports by watchdogs. Figure 2 depicts the flowchart of the TEDG algorithm.

In the following, we introduce the proposed algorithm in detail. The clustering, tree construction, watchdog selection, and trust evaluation schemes are discussed in Sects. 5.1 to 5.4.

5.1 Clustering

The first phase of the algorithm organizes nodes into clusters, where the aim is to adopt nodes with a higher trust level and residual energy as CHs. The PSO-TVAC algorithm is applied to perform clustering as well as tree construction and watchdog selection due to its high performance and adaption to our problem. The proposed clustering scheme comprises common components of PSO, including particle representation, population initialization, fitness function, and decoding procedure, which are described in the following.

5.1.1 Particle representation

Each particle P_i^C is an array of length $k \times n$. Dimension p_{ij}^C comprises two values that are chosen from ranges $[0 - l]$ and $[0 - w]$, respectively. These values are used to specify the j $^{\text{th}}$ corresponding CH to particle P_i^C, namely ch_i^j.

Figure 3 depicts a particle for an assumed WSN. The illustrated network in this figure is of dimensions 100 m × 100 m, and the left-down corner of the field is assumed to be the origin. It has 40 nodes, where five of them are adopted as CH. Accordingly, as shown in Fig. 3b, the particle is an array of dimensions 2 × 5.

Table 4 The values of parameters of PSO-TVAC

Parameter	Value
$w_{\max}$	0.9
$w_{\min}$	0.4
c_{1e}	0.5
c_{1s}	2.5
c_{2e}	2.5
c_{2s}	0.5

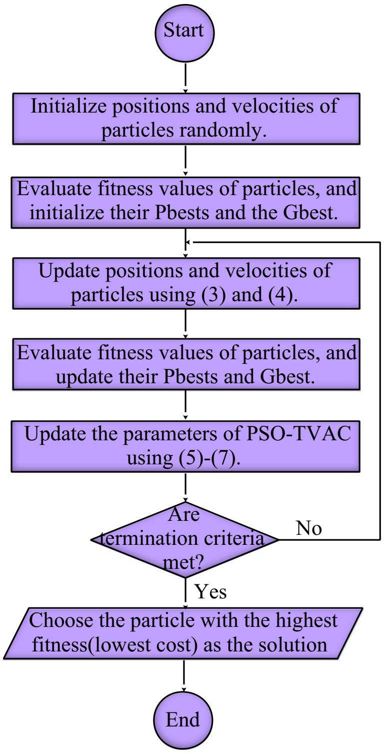

Start
Initialize positions and velocities of particles randomly.
Evaluate fitness values of particles, and initialize their Pbests and the Gbest.
Update positions and velocities of particles using (3) and (4).
Evaluate fitness values of particles, and update their Pbests and Gbest.
Update the parameters of PSO-TVAC using (5)-(7).
Are termination criteria met?
No
Yes
Choose the particle with the highest fitness(lowest cost) as the solution
End

Fig. 1 Flowchart of PSO-TVAC

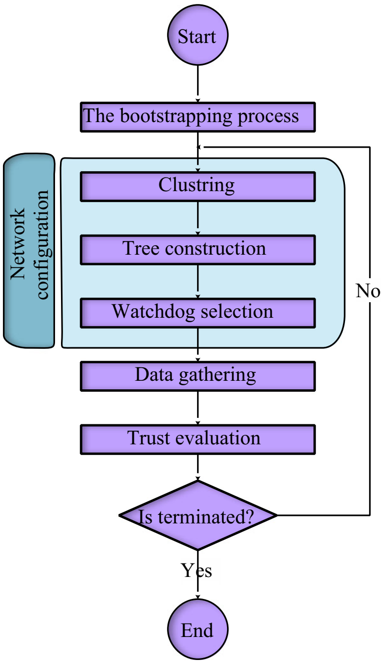


Fig. 2 Flowchart of TEDG

5.1.2 Initialization

The swarm comprises np^C particles. The dimensions of each particle are initialized to random values that are chosen from the specified ranges.

5.1.3 Fitness function

We intend to construct trust-aware and energy-efficient clusters. Regarding trustworthiness, the CHs should be trusty, so they do not drop sent data packets by the cluster members. Energy-efficient clustering necessitates adopting high-energy nodes as CHs and balancing the depleted energy by members for intra-cluster communication. According to the above discussion, we include three metrics in (8). The first two metrics are the average trust value and residual energy of CHs. Using these criteria results in choosing CHs with a high trust level and residual energy. The third measure, i.e., balancing the consumed energy by members, is stated as minimizing the standard deviation of distances of members from their corresponding CHs. The rationality behind this decision is that according to (2), the energy exhausted by each cluster member is proportional to its distance from the corresponding CH. Thus, the third criterion balances the amount of energy exhausted by cluster members for intra-cluster communication.

$$F_i^C = w_1 \mathrm{avg}_{1\le j\le k\times n}\left(t_{ch_i^j}\right) + w_2 \frac{\mathrm{avg}_{1\le j\le k\times n}\left(e_{ch_i^j}\right)}{e_{init}} + w_3 \frac{\left(1-\mathrm{std}_{s_k\in C_i^j}\left(d_{k,ch_i^j}\right)\right)}{tr}, w_1+w_2+w_3=1 \tag{8}$$

where F_i^C presents the fitness of particle P_i^C. Additionally, avg and std stand for average and standard deviation functions, respectively. Notation C_i^j denotes the j th corresponding cluster to the particle. Finally, w_1, w_2, and w_3 determine the impact of the above-mentioned measures.

5.1.4 Decoding procedure

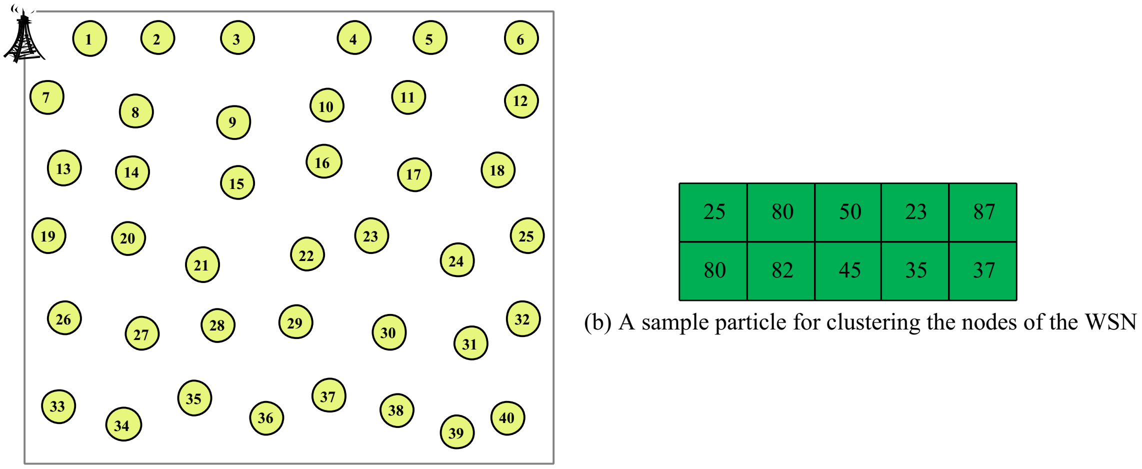
Particle P_i^C is decoded considering the values of its dimensions. Each dimension p_{ij}^C identifies the j th corresponding CH to the particle, namely ch_i^j. The two values of p_{ij}^C are assumed to be the coordinates of a point in the monitoring area. The closest node to this point is adopted as ch_i^j.

The remaining point is to determine the members of the clusters. Each non-CH node s_k is assigned to a CH that is located in its transmission range. The properness of $ch_i^j \in N_k$ to be adopted as the CH of s_k, namely pc_i^{jk}, is defined as:

$$pc_i^{jk} = t_{ch_i^j} + \frac{e_{ch_i^j}}{e_{init}} + \left(1-\frac{m_i^j}{\max\limits_{1\le q\le k\times n} m_i^q}\right) + \left(1-\frac{d_{k,ch_i^j}}{tr}\right) \tag{9}$$

where m_i^j denotes the number of members of C_i^j. As shown in (9), the criteria for CH selection are the trust value that the CH has gained, the amount of its residual energy, its degree value, and its distance from s_k. More precisely, it is preferred to adopt a CH with a higher trust value and residual energy, fewer members, and closer to the node.

The steps of decoding the presented particle in Fig. 3b are shown in Fig. 4. Firstly, the specified points by the particle are highlighted in Fig. 4a. Next, as depicted in Fig. 4b, the closest nodes to these points are adopted as CHs. The non-CH sensors are assigned to CHs using (9).



(a) An assumed WSN

25	80	50	23	87
80	82	45	35	37

(b) A sample particle for clustering the nodes of the WSN

Fig. 3 An example particle for the clustering phase

The flowchart of the proposed clustering scheme is presented in Fig. 5.

5.2 Tree construction

In this phase, a tree is constructed over the CHs and some relay nodes to gather data. Using this tree, each CH or relay node has a path to the sink. Additionally, each non-CH and non-relay node is connected to the tree via its corresponding CH. Accordingly, every sensor has a path for data transmission toward the sink. The proposed tree construction scheme aims to adopt sensors with a higher trust level and residual energy as relay nodes. This scheme increases data gathering reliability. The following describes particle representation, population initialization, fitness function, and decoding procedure for the proposed PSO-based scheme.

5.2.1 Particle representation

Each particle P_i^T is an array of length n, where the value of dimension p_{ij}^T is chosen from the range $[(-1)-1]$. As explained in the decoding procedure, these dimensions are used to construct a tree over the CHs and some relay nodes. Figure 6 exemplifies a particle for the tree construction problem over the proposed clustering scheme in Fig. 4b.

5.2.2 Initialization

The swarm consists of np^T particles. The dimensions per particle are chosen from the range $[(-1)-1]$ randomly.

5.2.3 Fitness function

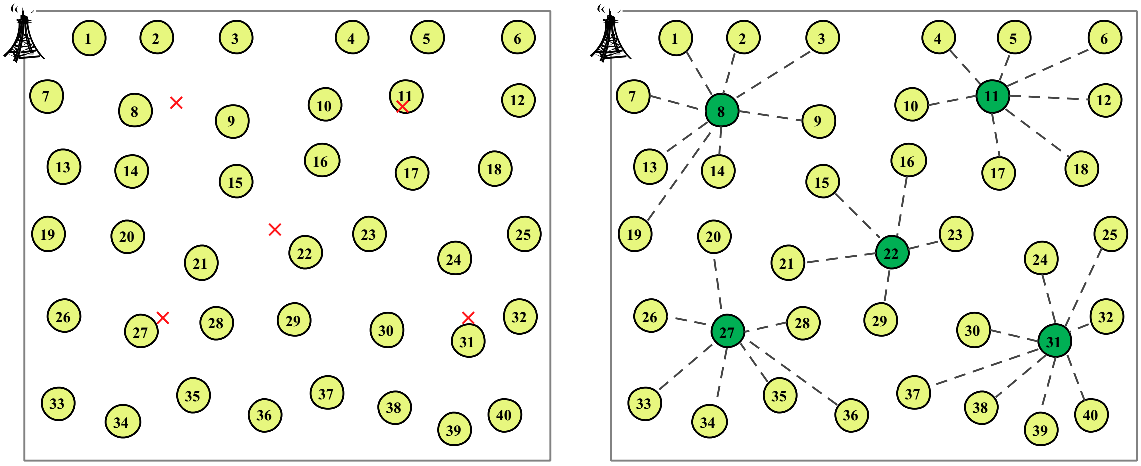
Three measures are used to evaluate the fitness of particle P_i^T, which is denoted by F_i^T. The considered measures are the trust level, residual energy, and the number of nodes on the tree. As shown in (10), the aim is to increase the average trust level and residual energy of the nodes, while minimizing the number of relay nodes used to construct the tree.

$$F_i^T = w_4 \text{avg}_{s_k \in T_i}(t_k) + w_5 \frac{\text{avg}_{s_k \in T_i}(e_k)}{e_{init}} + w_6\left(1 - \frac{nt_i}{n}\right), \quad w_4 + w_5 + w_6 = 1 \tag{10}$$

where T_i and nt_i represent the corresponding tree to P_i^T and the number of its nodes, respectively. Additionally, parameters w_4, w_5, and w_6, specify the importance of the considered criteria.

5.2.4 Decoding procedure

We modify the proposed scheme in Elhabyan and Yagoub (2015) for particle decoding. In this procedure, particle P_i^T is assumed to be of dimension n, where j $^{\text{th}}$ dimension demonstrates the priority of s_j to be included in T_i. The higher priority nodes are used for tree construction as described in the following. For each CH ch_j, a path connecting the CH to the sink is constructed. The construction of the path is initiated from the CH. The neighboring node of ch_j with the highest priority is adopted as its next hop toward the sink. This procedure is continued until the sink is reached. The data gathering tree is completed after connecting all CHs to the sink. The mentioned procedure yields constructing long paths and has a slow convergence



(a) The specified points by the particle

(b) The corresponding clustering scheme

Fig. 4 Decoding of the illustrated particle in Fig. 3b

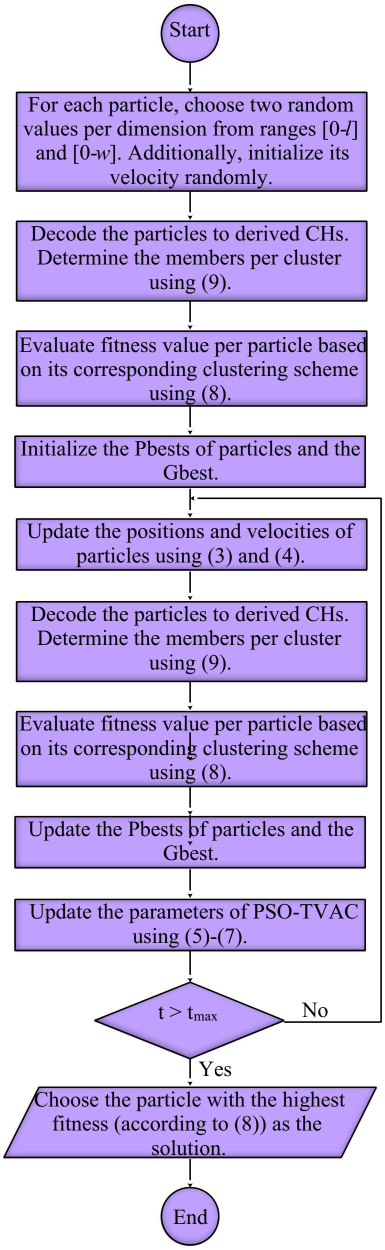

Start
For each particle, choose two random values per dimension from ranges [0-*l*] and [0-*w*]. Additionally, initialize its velocity randomly.
Decode the particles to derived CHs. Determine the members per cluster using (9).
Evaluate fitness value per particle based on its corresponding clustering scheme using (8).
Initialize the Pbests of particles and the Gbest.
Update the positions and velocities of particles using (3) and (4).
Decode the particles to derived CHs. Determine the members per cluster using (9).
Evaluate fitness value per particle based on its corresponding clustering scheme using (8).
Update the Pbests of particles and the Gbest.
Update the parameters of PSO-TVAC using (5)-(7).
t > t_{max}
No
Yes
Choose the particle with the highest fitness (according to (8)) as the solution.
End

Fig. 5 Flowchart of the clustering phase

speed. The reason is that the next hop of node s_k on an assumed path could be further from the sink than the node. To tackle this problem, we restrict the set of the allowed next hops of s_k to those neighbors that are closer to the sink. This strategy prevents the formation of long paths, and consequently, the proposed PSO-based scheme converges quickly.

Figure 7 illustrates the steps of decoding the represented particle in Fig. 6. As shown in this figure, the next hop of each node is one of its neighbors that is closer to the sink and has the highest priority. For example, consider node s_{22}. The neighbors of this sensor are nodes s_{15}, s_{16}, s_{21}, s_{23}, s_{29}, and s_{31}, which have priorities 0.82, 0.76, 0.17, -0.47, -0.67, and 0.72. Among these neighbors, nodes s_{15}, s_{16}, and s_{21} are closer to the sink. Considering the priorities of these nodes, sensor s_{15} is selected as the next hop of s_{22}.

Figure 8 illustrates the flowchart of the proposed tree construction scheme.

5.3 Watchdog selection

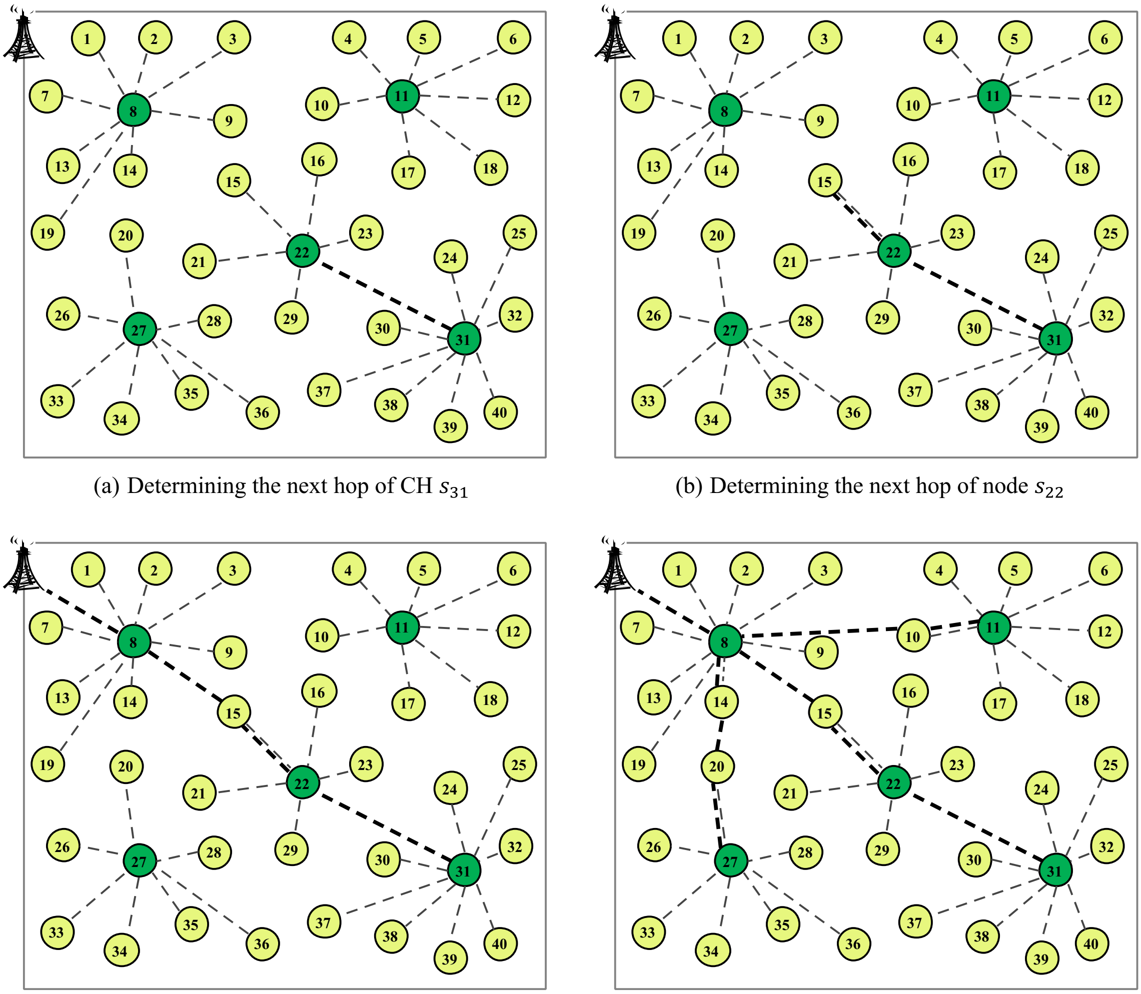
The last phase is devoted to selecting some watchdogs from non-CH and non-relay sensors. These nodes measure the data forwarding rate of their monitored ones and notify the sink about it. Finally, the sink estimates the trust value of nodes from the received reports and identifies malicious ones. It should be noted that similar to other sensors, the watchdog nodes monitor their surrounding environment and send their data to their corresponding CHs for further processing and routing toward the sink.

Some points are considered in the proposed watchdog selection algorithm. First, the nodes with a higher trust level and residual energy have priority to be chosen as watchdogs. The reports of highly trusted nodes are more likely to be accurate. Additionally, the adopted watchdogs should have enough energy to perform monitoring. The second point is that the malicious nodes should be prevented from being selected as watchdogs. A malicious node may act honestly to be adopted as a watchdog in future. This malicious watchdog can prepare false reports for the assigned nodes, and recognize them as malicious. The proposed solution to mitigate this problem is to adopt watchdogs randomly. The next point is to reduce the overhead of monitoring tasks. Watchdog assignment to all nodes increases the amount of exhausted energy throughout the WSN. The reason is that the watchdog nodes should be

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0.65	0.75	-0.16	0.61	0.56	-0.46	0.38	0.89	0.83	0.79	0.84	-0.33	-0.23	0.85	0.82	0.76	0.11	-0.52	0.80	0.83

21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
0.17	0.78	-0.47	-0.34	0.13	0.41	0.77	0.08	-0.67	0.56	0.72	-0.45	0.37	-0.61	0.28	-0.19	0.31	-0.61	0.43	0.12

Fig. 6 An example particle for tree construction over the depicted clustering scheme in Fig. 4b



(a) Determining the next hop of CH s_{31}

(b) Determining the next hop of node s_{22}

(c) Data transmission path from CH s_{31} to the sink

(d) The ultimate tree

Fig. 7 Some steps of decoding the depicted particle in Fig. 6

awake to perform monitoring tasks. Therefore, the watchdogs are specified only for CHs and relay nodes of the tree to balance the amount of exhausted energy and network protection. Non-CH nodes are monitored by their corresponding CHs. Finally, the assigned watchdog to each CH or relay node is one of its neighbors. Therefore, the allowed location of the watchdog per node is limited to its transmission range. Keeping these all together, a PSO-based watchdog selection scheme is proposed. The components of this algorithm are given below.

5.3.1 Particle representation

Each particle P_i^W is an array of length nw_i. Each dimension p_{ij}^W comprises two values, which are randomly adopted from the ranges $[0 - l]$ and $[0 - w]$. As discussed in the decoding procedure, each dimension p_{ij}^W is used to specify the j $^{\text{th}}$ corresponding watchdog to P_i^W. Accordingly, the number of watchdogs in the corresponding WSN to P_i^W becomes equal to nw_i. Figure 9 shows an example particle for the watchdog selection scheme. This particle is used to choose some watchdogs to monitor the specified CHs and relay nodes in Fig. 7d.

5.3.2 Initialization

The swarm includes np^W particles. Each particle P_i^W presents a solution to the problem, where at least a watchdog is assigned per CH and relay node. Accordingly, to construct P_i^W, firstly proper watchdogs are selected for monitoring tasks. Next, P_i^W is constructed considering the adopted watchdogs.

The proposed scheme adopts watchdogs in multiple rounds. In each round, the corresponding watchdog to a CH or relay node, namely s_j, is adopted. The set of candidate watchdogs of s_j, which comprises its non-CH and non-relay neighbors, is named as $\boldsymbol{CW_j}$. The reason that each node is at most given one role is to enhance security and balance energy consumption. To determine the watchdog of s_j in particle P_i^W, namely wd_i^j, low-trusted and low-energy nodes are removed from $\boldsymbol{CW_j}$ firstly. By low-trusted and low-energy nodes, we mean those sensors whose trust value and residual energy are less than the average. Next, the properness of each node $s_k \in \boldsymbol{CW_j}$ to be adopted as wd_i^j, namely pw_k, is computed as:

$$pw_k = \frac{e_k}{e_{init}} + t_k \tag{11}$$

The probability of selecting node s_k as wd_i^j, denoted by pr_j^k, is computed using (12). Finally, wd_i^j is adopted using Roulette-Wheel Selection (RWS). The application of RWS prevents the algorithm from trapping into local optimums and results in more reasonable solutions.

$$pr_j^k = \frac{pw_k}{\sum_{s_q \in \boldsymbol{CW_j}} pw_q} \tag{12}$$

The pseudo-code of the proposed scheme is given in Algorithm 1. In this context, $\boldsymbol{WD_i}$ denotes the corresponding watchdog set to P_i^W. Additionally, $\boldsymbol{M}$ is the set of monitored nodes and comprises CHs and relay nodes. To initialize P_i^W, set $\boldsymbol{WD_i}$ is formed first. After that, the coordinates of the watchdogs are used to form the particle.

watchdogs. The aim is to choose the least number of high-energy and trusted nodes as watchdogs to effectively monitor CHs and relay nodes. The fitness of particle P_i^W, namely F_i^W, is computed as:

$$F_i^W = w_7 \mathrm{avg}_{s_k \in \boldsymbol{P_i^W}}(t_k) + w_8 \frac{\mathrm{avg}_{s_k \in \boldsymbol{P_i^W}}(e_k)}{e_{init}} + w_9\left(1 - \frac{nw_i}{n}\right), w_7 + w_8 + w_9 = 1 \tag{13}$$

where coefficients w_7, w_8, and w_9, are used to specify the impact of the measures.

5.3.4 Decoding procedure

The set of watchdog nodes is derived by decoding particle P_i^W. Each dimension p_{ij}^W comprises two values, which specify the coordinates of a point on the field. The closest node to this point is considered as the j th watchdog. The decoding steps of the represented particle in Fig. 9 are shown in Fig. 11. Firstly, as illustrated in Fig. 9a, the corresponding points to the particle are specified in the monitoring area. Next, the closest nodes to these points are selected as watchdogs (Fig. 11b).

Algorithm 1 Initialization of particle $\boldsymbol{P_i^W}$

```
Input: WSN, set of clusters, data gathering tree.
Output: $P_i^W$.

$\boldsymbol{WD_i} \leftarrow \emptyset$.
$\boldsymbol{M} \leftarrow \{CHs\} \cup \{relay\ nodes\}$.
for each node $s_j \in \boldsymbol{M}$ do
    $\boldsymbol{CW_j} \leftarrow \{s_k | s_k \in \boldsymbol{N_j}\ \&\ s_k \notin \boldsymbol{M}\ \&\&\ \mathrm{avg}_{1 \le q \le n}(t_q) \le t_k\ \&\&\ \mathrm{avg}_{1 \le q \le n}(e_q) \le e_k\}$.
    For each node $s_k \in \boldsymbol{CW_j}$, compute $pw_k$ using (11).
    For each node $s_k \in \boldsymbol{CW_j}$, compute $pr_j^k$ using (12).
    Adopt $wd_i^j$ using the RWS mechanism.
    $\boldsymbol{WD_i} \leftarrow \boldsymbol{WD_i} \cup \{wd_i^j\}$
end for
for each dimension $p_{ij}^W$ do
    $p_{ij}^W$ ←Coordinates of the $j^{\text{th}}$ watchdog.
end for
```

An example watchdog set for the given network in Fig. 7d is represented in Fig. 10a. Additionally, the initial form of the particle is shown in Fig. 10b.

5.3.3 Fitness function

The considered measures in this phase are the average trust level, average residual energy, and the number of

5.3.5 Particle updating

As previously mentioned in Sect. 4, in PSO-TVAC, each particle P_i^W is updated based on its pbest and the gbest. The proposed formulas for particle updating, i.e., (3) and (4), implicitly assume that P_i^W**,** its pbest ($\boldsymbol{pbest_i^W}$), and the gbest ($\boldsymbol{gbest^W}$) have the same dimensions. More precisely, p_{ij}^W is updated using the j th dimension of $\boldsymbol{pbest_i^W}$ and $\boldsymbol{gbest^W}$. The

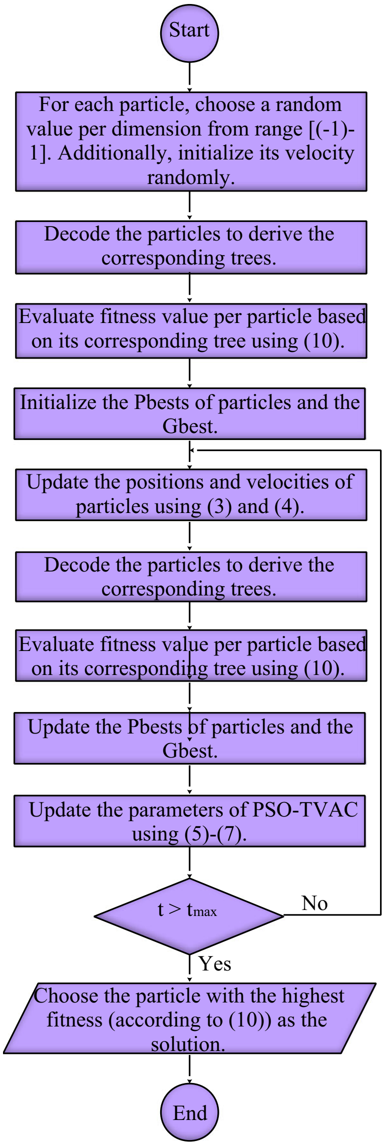

Start
For each particle, choose a random value per dimension from range [(-1)-1]. Additionally, initialize its velocity randomly.
Decode the particles to derive the corresponding trees.
Evaluate fitness value per particle based on its corresponding tree using (10).
Initialize the Pbests of particles and the Gbest.
Update the positions and velocities of particles using (3) and (4).
Decode the particles to derive the corresponding trees.
Evaluate fitness value per particle based on its corresponding tree using (10).
Update the Pbests of particles and the Gbest.
Update the parameters of PSO-TVAC using (5)-(7).
t > tmax
No
Yes
Choose the particle with the highest fitness (according to (10)) as the solution.
End

Fig. 8 Flowchart of the tree construction phase

issue raises here is that in the watchdog selection algorithm, the particles are of different dimensions. Therefore, generally, there is no one-to-one correspondence between the dimensions of P_i^W, $\boldsymbol{pbest}_i^W$, and $\boldsymbol{gbest}^W$. To deal with this issue, we consider two cases based on the number of dimensions of P_i^W, $\boldsymbol{pbest}_i^W$, and $\boldsymbol{gbest}^W$. The solution is stated for $\boldsymbol{pbest}_i^W$, which is assumed to be of dimension nb_i. The same approach is applied to handle $\boldsymbol{gbest}^W$.

45	53	5	75
95	67	50	37

Fig. 9 An example particle for watchdog selection over the illustrated WSN in Fig. 7d

1. P_i^W has less or equal dimensions compared to $\boldsymbol{pbest}_i^W$. In this case, p_{ij}^W corresponds to dimension j of the pbest as usual. Additionally, the residual dimensions of $\boldsymbol{pbest}_i^W$ are ignored.
2. P_i^W has more dimensions than $\boldsymbol{pbest}_i^W$. In this case, for the first nb_i dimensions, dimension j of P_i^W corresponds to that of $\boldsymbol{pbest}_i^W$. The corresponding dimension for $p_{ij}^W\,(nb_i < j \le nw_i)$ is determined as follows. Firstly, all dimensions of $\boldsymbol{pbest}_i^W$ are decoded to obtain some points on the monitoring area. The dimension p_{ij}^W is also decoded to derive a point on the area, namely pt_{ij}. That dimension of $\boldsymbol{pbest}_i^W$, which its corresponded point has the least distance from pt_{ij}, is adopted as the corresponding dimension to p_{ij}^W.

Algorithm 2 presents the steps of the proposed particle updating scheme more precisely. In this algorithm, pb_{ij}^W and gb_{ij}^W denote the corresponding dimension to p_{ij}^W in $\boldsymbol{pbest}_i^W$ and $\boldsymbol{gbest}^W$, respectively.

The proposed particle updating scheme is more clarified in the following example. Here, the depicted particle in Fig. 9 is considered as P_i^W. Additionally, $\boldsymbol{pbest}_i^W$ and $\boldsymbol{gbest}^W$ are given in Figs. 12a,b, respectively. Furthermore, their corresponding WSNs are shown in Figs. 12c,d for more clarity. The pbest is of dimension four and handled as usual. On the other hand, the gbest is of dimension three and therefore, pg_{i4}^W is chosen from the three dimensions of $\boldsymbol{gbest}^W$. As derived from Fig. 12b, the corresponding point to the third dimension of $\boldsymbol{gbest}^W$ has the least distance from pt_{ij}. Accordingly, the third dimension of $\boldsymbol{gbest}^W$ is chosen as pg_{i4}^W.

Algorithm 2 Particle updating of watchdog selection

Input: P_i^W, $\boldsymbol{pbest_i^W}$, $\boldsymbol{gbest^W}$.

Output: Updated P_i^W.

```
for each dimension $p_{ij}^W$ do
    if $j \le nb_i$ do
        $pb_{ij}^W \leftarrow$Dimension $j$ of $\boldsymbol{pbest_i^W}$.
        $gb_{ij}^W \leftarrow$Dimension $j$ of $\boldsymbol{gbest^W}$.
    else
        $cb \leftarrow$The closest point to $pt_{ij}$, from the points derived by decoding $\boldsymbol{pbest_i^W}$.
        $pb_{ij}^W \leftarrow$The corresponding dimension to $cb$.
        $cg \leftarrow$The closest point to $pt_{ij}$, from the points derived by decoding $\boldsymbol{gbest^W}$.
        $gb_{ij}^W \leftarrow$The corresponding dimension to $cg$.
    end if
end for

Update $P_i^W$ using (3) and (4).
```

The flowchart of the proposed watchdog selection scheme is given in Fig. 13.

5.4 Trust evaluation

The sink updates the trust values of the sensor nodes at the end of each round. To this end, the moving average model is applied, which is presented in the following formula:

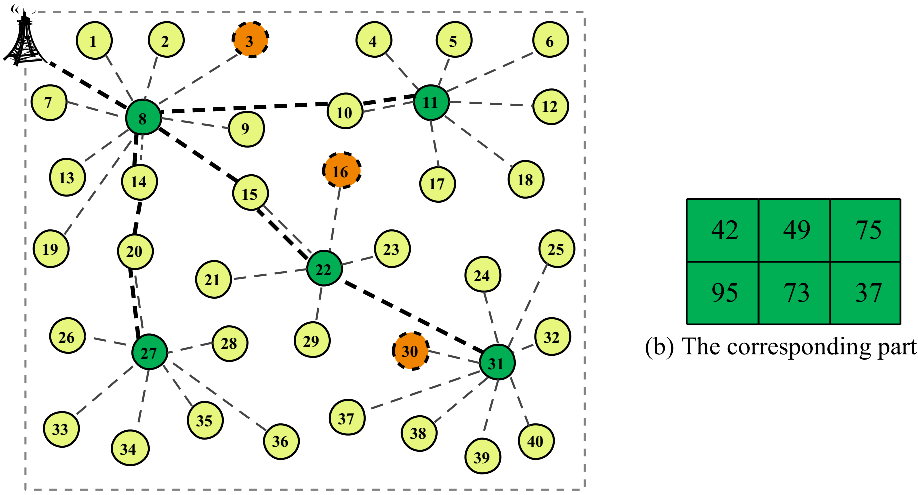
$$t_i = w_{10}t_i + (1 - w_{10})tr_i \tag{14}$$

where tr_i denotes the trust value of node s_i in the current round. This parameter is derived considering the evidence gathered from the monitors (i.e., watchdogs or the corresponding CH) of sensor s_i. Furthermore, parameter w_{10} determines the importance of t_i against the gained trust by s_i in the current round.

In the TEDG algorithm, the monitoring scheme for each node depends on its role. The CHs and relay nodes are assigned watchdogs. The ordinary sensors, which only transmit the sensed data to the CHs, have no special monitors. The corresponding CH to each ordinary node is responsible for monitoring its behavior and reporting its data generation rate to the sink. As previously mentioned, the reason for applying different policies based on the node type is to balance the security level and the energy consumption criterion. According to the above discussion, the trust of ordinary node $s_i \in \boldsymbol{C_j}$ in the current round is computed as:

$$tr_i = t_{ch_j} \times td^i_{ch_j} \tag{15}$$

In this equation, the trust value of node ch_j is also considered to compute tr_i. This strategy alleviates the impact of false reports of malicious CHs on the trust values

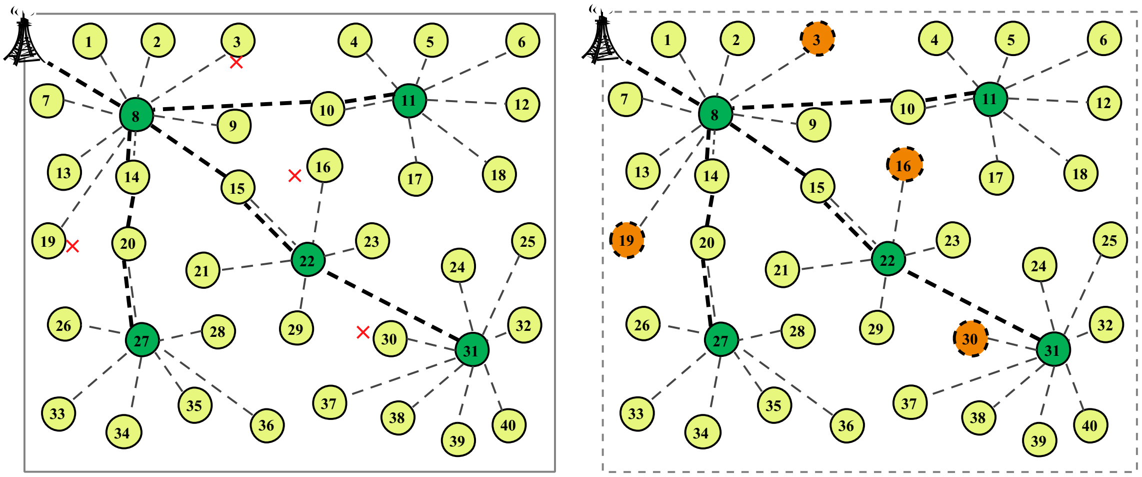


(a) The selected watchdogs, which are illustrated by dashed circles

42	49	75
95	73	37

(b) The corresponding particle

Fig. 10 Watchdog selection over the depicted WSN in Fig. 7d



(a) The specified points by the particle

(b) The corresponding watchdogs, which are presented by dashed circles

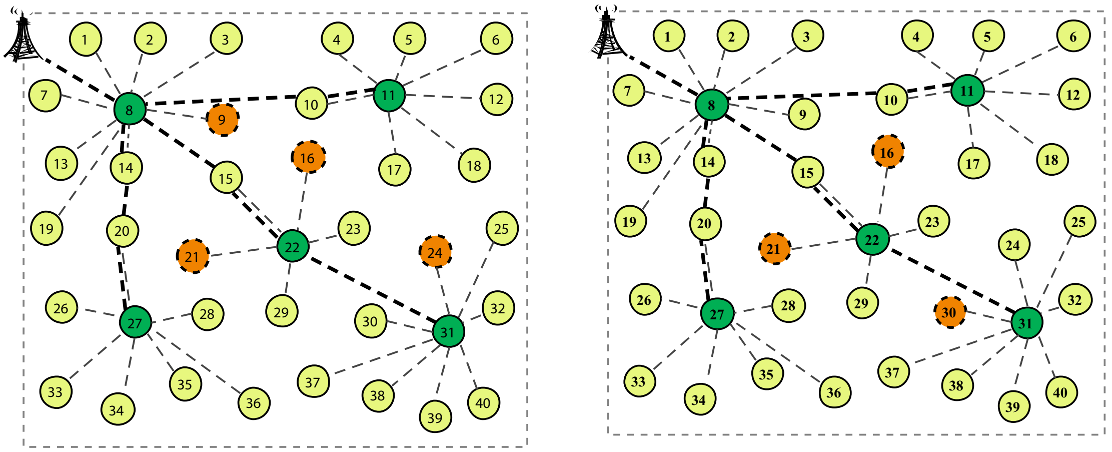
Fig. 11 Decoding of the depicted particle in Fig. 9

45	56	38	80
75	63	40	41

(a) The pbest

50	39	73
71	42	39

(b) The gbest



(c) The corresponding watchdogs to the particle of Fig. 12(a)

(d) The corresponding watchdogs to the particle of Fig. 12(b)

Fig. 12 The pbest and gbest of the presented particle in Fig. 9

of truthful nodes. The notation $td^i_{ch_j}$ denotes the trust degree of ch_j in node s_i in the current round. Node ch_j computes this parameter according to its direct observations of packet generation by node s_i in the current round as:

$$td^i_{ch_j} = \frac{NGP_i}{EGP} \tag{16}$$

where NGP_i denotes the number of generated packets by node s_i at the current round. Additionally, EGP presents the expected number of generated packets at the current round, which is the same for all nodes over time as we assume periodic data generation.

The trust value of node s_i, which is a CH or relay node, is derived based on the given evidence by the watchdogs in its neighboring set as:

$$tr_i = \frac{\sum_{j=1}^{nd_i} t_j \times td^i_{wg_i^j}}{nd_i} \tag{17}$$

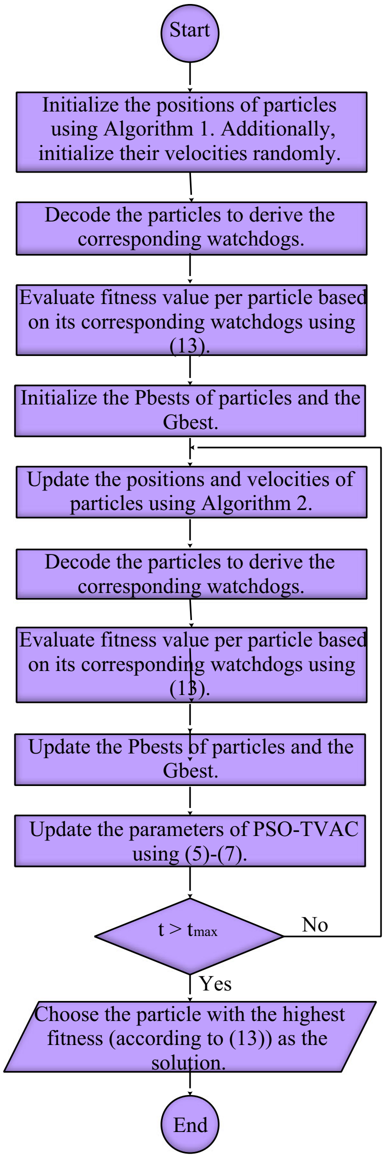

Start
Initialize the positions of particles using Algorithm 1. Additionally, initialize their velocities randomly.
Decode the particles to derive the corresponding watchdogs.
Evaluate fitness value per particle based on its corresponding watchdogs using (13).
Initialize the Pbests of particles and the Gbest.
Update the positions and velocities of particles using Algorithm 2.
Decode the particles to derive the corresponding watchdogs.
Evaluate fitness value per particle based on its corresponding watchdogs using (13).
Update the Pbests of particles and the Gbest.
Update the parameters of PSO-TVAC using (5)-(7).
t > tmax
No
Yes
Choose the particle with the highest fitness (according to (13)) as the solution.
End

Fig. 13 Flowchart of the watchdog selection phase

where nd_i and wg_i^j denote the number of available watchdogs in $\boldsymbol{N_i}$, and the j th watchdog of sensor node s_i, respectively. It should be noted that in the proposed watchdog selection scheme, only one watchdog is assigned per CH or relay node. However, the determined watchdog for a CH or relay node may fall in the transmission range of others. As a result, each CH or relay node may be monitored by multiple watchdogs.

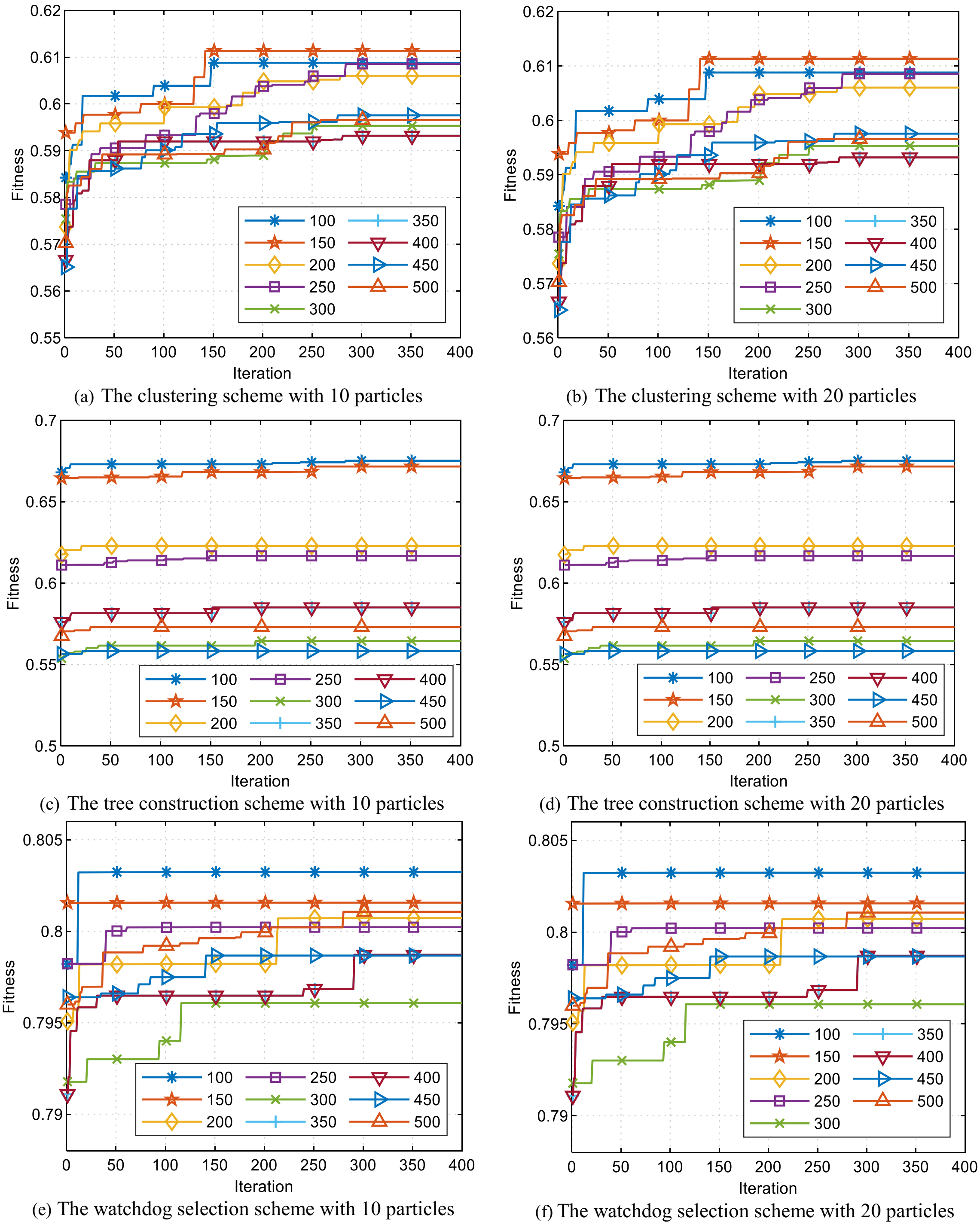
6 Performance evaluation

The performance of TEDG is studied in this section. Some recently published algorithms, including TPSO-CR (Elhabyan and Yagoub 2015), DAMS (Abdellatif and Mosbah 2020), TEFCSRP (Mittal et al. 2021), and CAT-EDP (Shahid et al. 2022) are adopted for the sake of comparison. TPSO-CR applied PSO for clustering and tree construction. This algorithm did not concern SFA. On the other hand, DAMS, TEFCSRP, and CAT-EDP are trust-aware algorithms. DAMS partitioned nodes into clusters, where CHs send data directly to the sink. The TEFCSRP algorithm used cuckoo search and fuzzy logic for cluster construction. Additionally, it proposed a greedy scheme for tree construction. Finally, CAT-EDP applied cellular automata for clustering. The comparisons among the considered algorithms are reported in Figs. 14, 15, 16, 17, 18, 19, 20, 21, 22, 23. Each data point in these figures is the average of five experiments over randomly deployed WSNs. Additionally, each experiment of AI-based schemes is repeated five times. Accordingly, each result for AI-based algorithms represents the average of 25 different executions of the algorithms. Furthermore, MATLAB is used to implement the considered algorithms. The comparison measures are packet loss, the exhausted energy by the nodes, and the network lifetime.

The simulations are conducted in areas of dimensions 100 m × 100 m to 200 m × 200 m. The number of sensors is varied within the range of [100–500]. The sensor nodes are deployed randomly over the network. Parameters tr and e_{init} are set to 60 m and 2 J, respectively. The value of γ is chosen from set $\{10\%, 20\%\}$. Accordingly, 10% or 20% percentage of sensors are randomly chosen as malicious nodes. The malicious sensors are randomly adopted from the deployed ones. Furthermore, the value of α, which presents the packet dropping probability of malicious nodes, is adopted from set $\{20\%, 30\%\}$. Finally, the value of k is assumed to be 10%. This value results in constructing clusters of acceptable sizes. The remaining point is to determine the values of parameters w_1 to w_{10}. The first nine parameters balance trustworthiness and energy efficiency in different phases. These parameters are determined such that the resulting packet loss, energy exhaustion, and network lifetime become acceptable. Based on the simulation results, these parameters are set to 0.3, 0.4, 0.3, 0.4, 0.35, 0.25, 0.5, 0.25, and 0.25. The last parameter, i.e., w_{10}, determines the impact of the past behavior of sensors on their trust value. This parameter is set to 0.8 in the performed simulations.

6.1 Determining the control parameters

The control parameters of the proposed algorithm can be divided into two sets. The first set includes the introduced parameters in PSO-TVAC, namely w_{max}, w_{min}, c_{1e}, c_{1s}, c_{2e}, and c_{2s}. The values of these parameters are assumed to be 0.9, 0.4, 0.5, 2.5, 2.5, and 0.5, as stated in Ratnaweera et al. (2004). The next set consists of the number of iterations



(a) The clustering scheme with 10 particles

(b) The clustering scheme with 20 particles

(c) The tree construction scheme with 10 particles

(d) The tree construction scheme with 20 particles

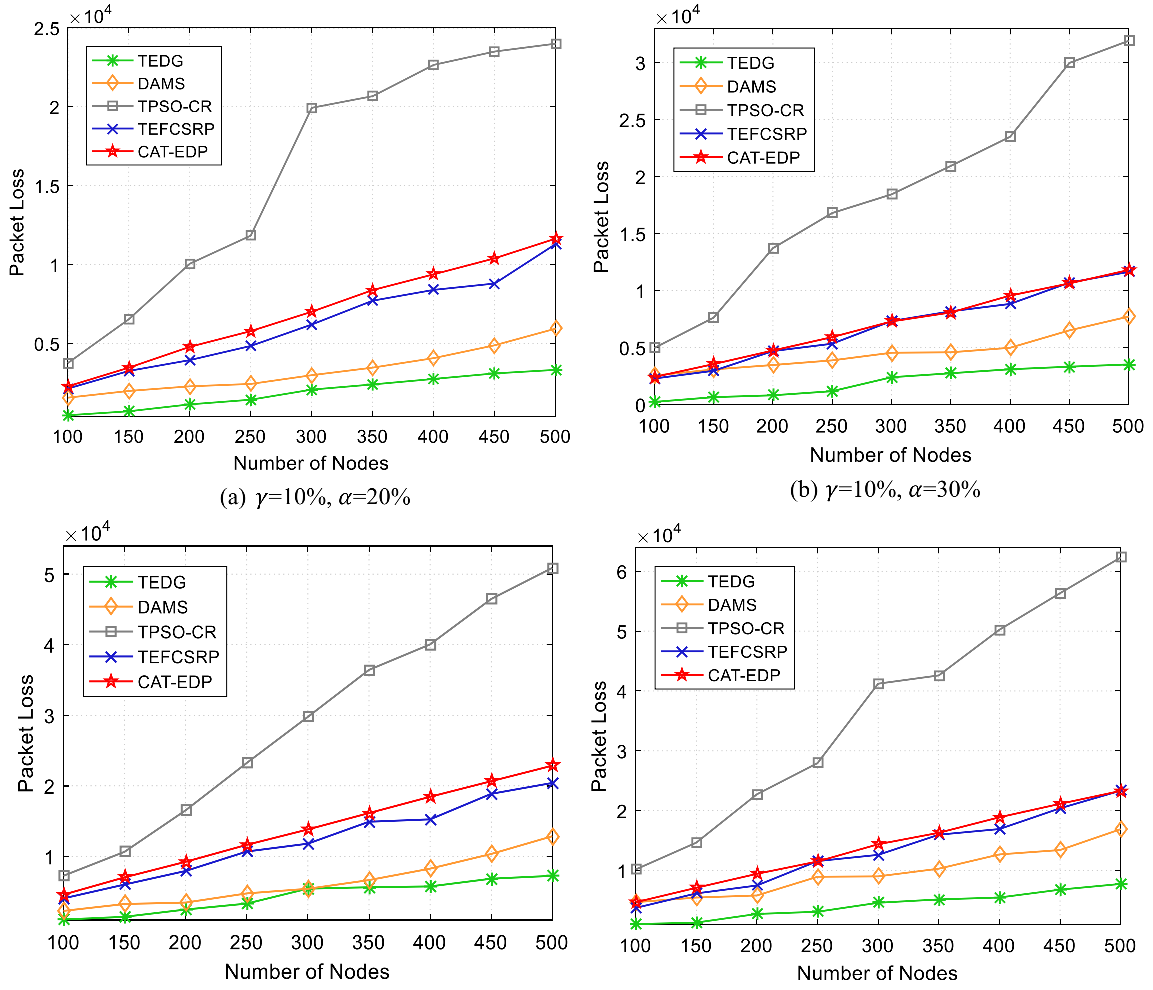
(e) The watchdog selection scheme with 10 particles

(f) The watchdog selection scheme with 20 particles

Fig. 14 Convergence plot of the phases of TEDG versus different numbers of nodes and particles

and particles of clustering, tree construction, and watchdog selection phases. We derive the convergence plot of these phases against different numbers of particles and nodes, and determine the proper number of iterations and particles per phase accordingly. In this set of experiments, parameters γ and α are set to 10% and 30%, respectively.

Figure 14 illustrates the convergence plot of clustering, tree construction, and watchdog selection phases versus different numbers of particles and sensor nodes. From this figure, we can see that the derived fitness values using 10 and 20 particles do not differ noticeably. Therefore, the number of particles is set to 10 to decrease the time complexity of the algorithm. Additionally, based on the



(a) γ=10%, α=20%

(b) γ=10%, α=30%

(c) γ=20%, α=20%

(d) γ=20%, α=30%

Fig. 15 Packet loss comparison versus different numbers of nodes and various values of γ and α

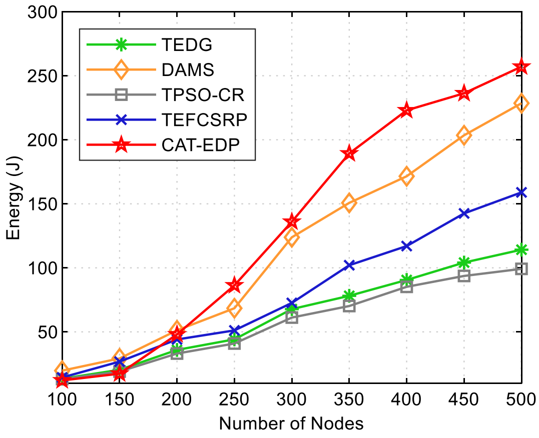


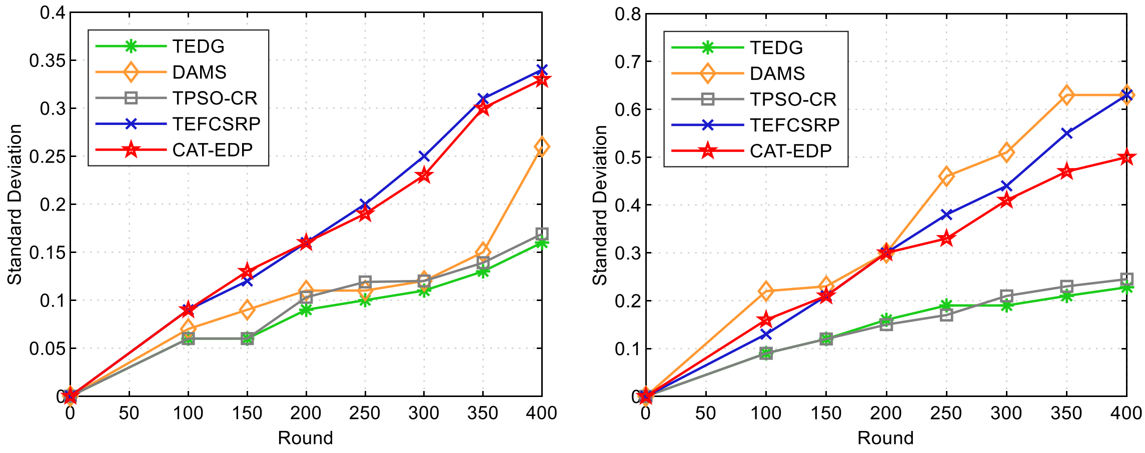
Fig. 16 Energy exhaustion comparison versus different numbers of nodes

presented results in Fig. 14a, c, and e, the maximum number of iterations (i.e., t_{max}) in all phases is assumed to be 300.

6.2 Packet loss comparison

The packet loss measure indicates the ability of the algorithms to prevent malicious nodes from disrupting the data gathering process. The packet loss of node s_i, namely pl_i, depends on the number of malicious nodes on the path connecting the node to the sink. Assume there are nm_i malicious nodes on the path from s_i to the sink, where the j th one is denoted by ml_i^j. The first malicious node, ml_i^1, drops α percentage of packets, and forwards the remaining ones. Node ml_i^2 drops α percentage of the received packets, which is equal to $\alpha(1-\alpha)$ percentage of the generated packets by s_i. Generally, ml_i^j drops $\alpha(1-\alpha)^{j-1}$ percentage of the generated packets by s_i. Accordingly, pl_i is equal to:

$$pl_i = \sum_{j=1}^{nm_i} \alpha(1-\alpha)^{j-1} \tag{18}$$



(a) 250-node WSNs

(b) 500-node WSNs

Fig. 17 Standard deviation (SD) of residual energy of nodes comparison over time

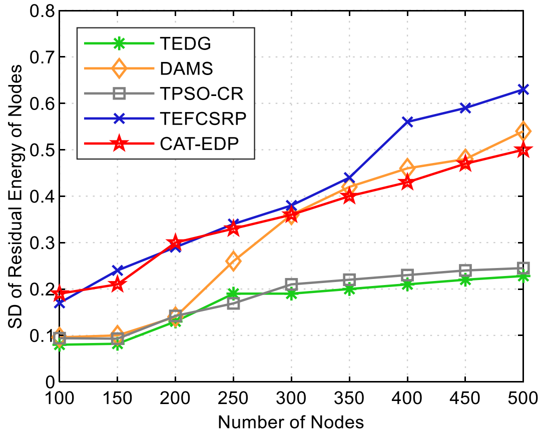


Fig. 18 SD of residual energy of nodes comparison versus different numbers of nodes

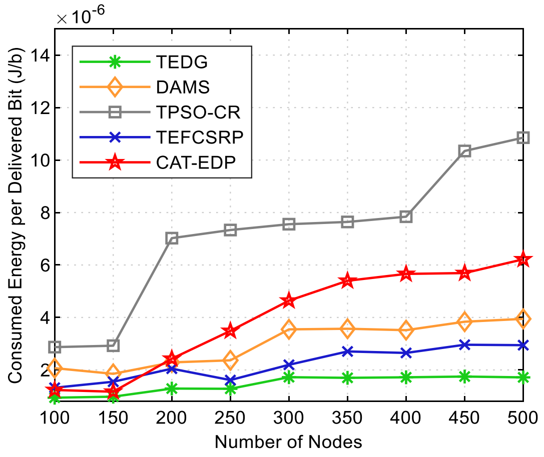


Fig. 19 Energy exhaustion per delivered bit comparison versus different numbers of nodes

Using the equation above, the total packet loss of the WSN will be:

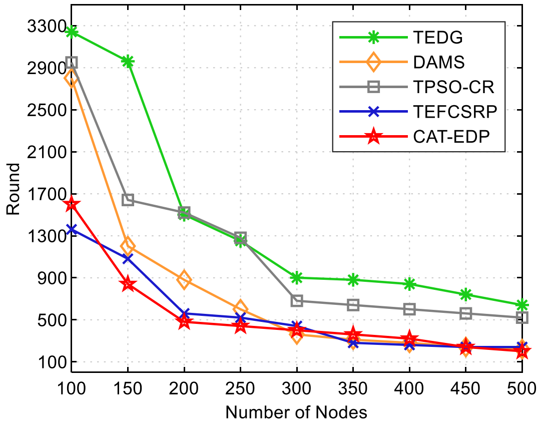
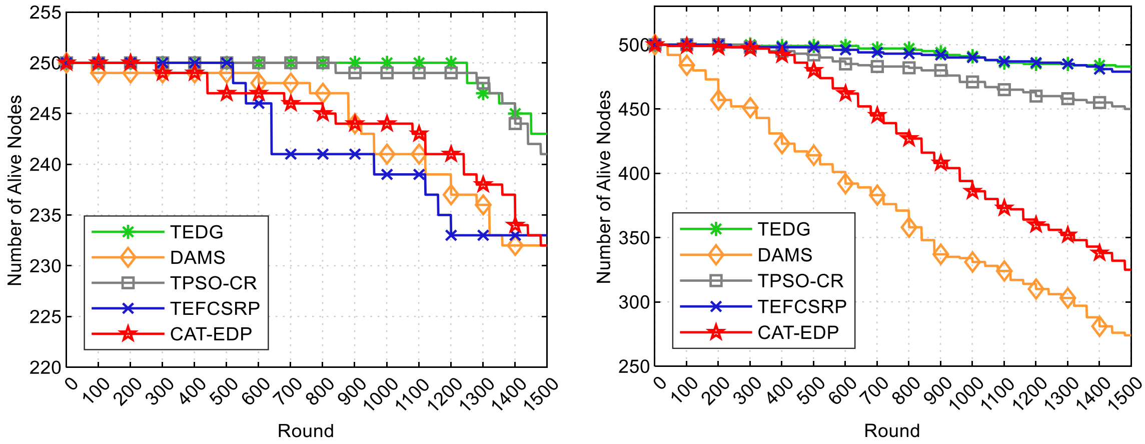


Fig. 20 Network lifetime comparison versus different numbers of nodes

$$\text{packet loss} = \sum_{i=1}^{n} pl_i = \sum_{i=1}^{n} \sum_{j=1}^{nm_i} \alpha(1-\alpha)^{j-1} \quad (19)$$

As it is derived from the above equation, the packet loss depends on the value of α and the number of malicious nodes that are chosen as relay nodes. Therefore, adopting highly trusted nodes as CHs and relay nodes reduces packet loss considerably.

The SFA mitigation capability of the considered algorithms versus various numbers of nodes is compared in Fig. 15. This criterion is measured by varying the percentage of malicious nodes, the packet dropping probability, and the number of nodes. As shown in this figure, the resultant packet loss of all algorithms increases by raising the number of nodes. Additionally, the average increase in packet loss caused by increasing γ from 10 to 20% is equal to 95%. Finally, increasing the percentage of dropped packets from 20 to 30% raises the packet loss by 16.7% on average.



(a) 250-node WSNs (b) 500-node WSNs

Fig. 21 Number of alive nodes comparison over time

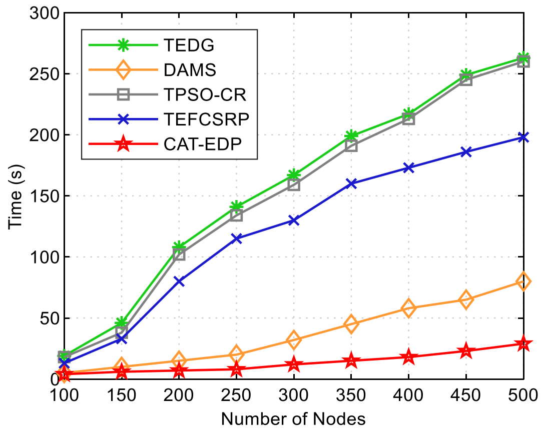


Fig. 22 Time Complexity comparison versus different number of nodes

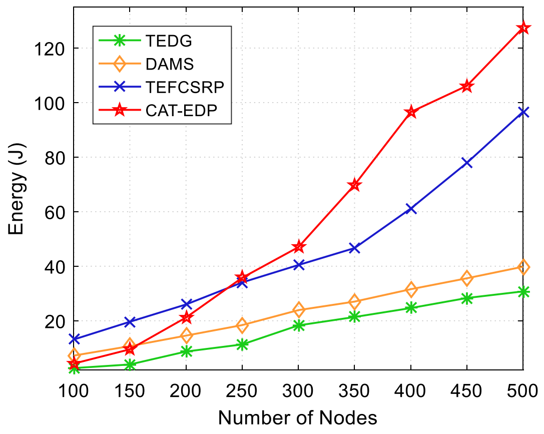


Fig. 23 Overhead comparison versus different number of nodes

According to the reported results in Fig. 15, TEDG improves the average packet loss by 7.99, 1.91, 3.08, 3.36 times, compared to TPSO-CR, DAMS, TEFCSRP, and CAT-EDP, respectively. TPSO-CR did not include an SFA mitigation mechanism. Therefore, malicious nodes may participate in the data gathering process. Adopting these nodes as CHs or relay nodes increases packet loss considerably. Therefore, the algorithm has a low performance despite achieving an acceptable amount of energy consumption and lifetime. DAMS is in second place regarding the packet loss criterion. In this algorithm, the CHs directly send data to the sink. Therefore, using DAMS decreases the probability of participating malicious nodes in the data gathering process. However, as shown in Sects. 6.3 and 6.4, this algorithm yields a high energy consumption and low network lifetime. The TEFCSRP algorithm monitored all sensors by their neighbors. Additionally, in CAT-EDP, each sensor is monitored by the nodes in its cluster. These algorithms did not concern the trustworthiness of watchdogs (i.e., neighbors and nodes in the cluster). Therefore, they dropped more packets compared to DAMS. The low packet loss of the proposed algorithm is due to applying PSO-TVAC in clustering, tree construction, and watchdog selection phases. The proper particle representation in these phases yields acceptable results. Additionally, the trust measure is included in the fitness functions of all phases. Adopting trusted nodes as CHs, relay nodes, and watchdogs reduces packet loss substantially.

6.3 Energy exhaustion comparison

Three energy-aware metrics are considered, including the average and standard deviation of the exhausted energy by the nodes, and the consumed energy per delivered bit to the sink. The average energy exhaustion of nodes indicates the total depleted energy for data gathering. Furthermore, the standard deviation of the residual energy of sensors provides evidence about the evenness of energy consumption throughout the network. The mentioned parameters, however, do not demonstrate the energy efficiency of the algorithms. An algorithm may have a low energy consumption, but its packet loss is high. Therefore, we also measure the consumed energy per delivered bit. This measure is derived by dividing the consumed energy of all nodes by the number of delivered bits to the sink.

Figure 16 illustrates the average energy consumption by varying the number of nodes. The reported results are derived after 400 rounds of executing the algorithms. In this section and the following experiments, parameters γ and α are assumed to be 10% and 30%, respectively. As shown in the figure, the consumed energy increases by enlarging the node set. TEDG reduces the average energy exhaustion by 28%, 84%, and 134%, in comparison with TEFCSRP, DAMS, and CAT-EDP, respectively. The higher energy exhaustion of DAMS and CAT-EDP is due to direct data transmission from CHs to the sink. According to (1), the long distances between the CHs and the sink necessitate a high energy consumption for data transmission. On the contrary, the CHs are connected to the sink using multi-hop paths in the proposed algorithm. Sending data over short distances does not require much energy. The TEFCSRP algorithm also used multi-hop paths for data transmission. However, it used only CHs in the tree resulting in relatively long distances between CHs, which yields high energy consumption for data transmission. Additionally, DAMS is greedy and consequently has a low performance. Finally, as shown in Sect. 6.6, TEFCSRP and CAT-EDP have a relatively high overhead, which increases their total energy consumption. The proposed algorithm also consumes 9% more energy than TPSO-CR. The lower energy consumption of this scheme is due to its higher packet loss. The lost packets are transmitted over fewer hops compared to the delivered ones to the sink. Accordingly, energy consumption decreases by increasing packet loss.

The standard deviation of residual energy of the sensor nodes over time using the competitive algorithms is shown in Fig. 17. According to the reported results in this figure, TEDG yields the least standard deviation. After 400 rounds of executing the algorithm, this measure raises to 0.16 and 0.23 for 250 and 500-node WSNs, respectively. The average of this metric for 250 and 500-node WSNs using other considered algorithms is equal to 0.23 and 0.58, respectively. The better outcome of TEDG is due to that it considers the residual energy of CHs in the cluster construction phase. Additionally, high-energy sensors are adopted as relay nodes and watchdogs in the successive phases of the algorithm. Finally, data is transmitted in a multi-hop manner toward the sink. This scheme avoids quick energy depletion of CHs. Accordingly, the sensor nodes exhaust energy more evenly by exploiting TEDG.

The impact of varying the number of nodes on the standard deviation criterion is depicted in Fig. 18. As shown in the figure, this measure increases by enlarging the node set. Additionally, the standard deviation of the proposed algorithm is less than other schemes. More precisely, compared to TPSO-CR, DAMS, TEFCSRP, and CAT-EDP, it reduces this measure by 5%, 73%, 137%, and 108%, respectively. The high standard deviation of DAMS and CAT-EDP is due to that the CHs directly send gathered data to the sink in these algorithms. Accordingly, the adopted CHs per round consume a considerable amount of energy. Furthermore, DAMS is greedy and therefore, it cannot balance the consumed energy by nodes effectively. Finally, TEFCSRP did not consider the energy criterion for adopting relay nodes on the data gathering tree. Accordingly, the algorithm has a high standard deviation.

The last studied energy-aware criterion is the amount of exhausted energy to deliver one bit to the sink. This metric is investigated in Fig. 19, where the data points are computed by dividing the consumed energy of all nodes by the number of delivered bits to the sink. As illustrated in this figure, the proposed algorithm yields the lowest amount of exhausted energy to deliver per bit and is the most energy-efficient scheme in our experiments. More specifically, the amount of exhausted energy by TEDG to deliver one bit to the sink is 1.46 μJ. This measure is equal to 7.15 μJ, 3 μJ, 2.22 μJ, and 3.97 μJ, for TPSO-CR, DAMS, TEFCSRP, and CAT-EDP, respectively. The superiority of our algorithm regarding the exhausted energy to deliver per bit measure is due to its lowest energy exhaustion and packet loss. According to the reported results in this section, the proposed algorithm outperforms the existing approaches regarding the energy consumption criterion.

6.4 Lifetime comparison

An important concern in WSNs is to decrease the death rate of sensors over time. This is due to that the coverage and connectivity of nodes may be distrusted by increasing the number of dead nodes. The death rate is quantified in the literature as the network lifetime and the number of alive nodes measures. The lifetime measure indicates the timespan between the network starting time and the first death. The second metric is defined as the number of alive nodes over time. These criteria are studied in Figs. 20 and 21, respectively. According to Fig. 20, the first node death in 250 and 500-node WSNs using the proposed algorithm occurs on rounds 1251 and 641, respectively. These values are equal to 1281 and 521 for the second-place algorithm, namely TPSO-CR. On average, TEDG increases network lifetime by 28%, 141%, 159%, and 187%, compared to TPSO-CR, DAMS, TEFCSRP, and CAT-EDP, respectively. The other point is that the network lifetime decreases by enlarging the node set. This is due to that by increasing the number of sensors, the closer nodes to the sink have to forward more data. These nodes consume more energy and die sooner.

Figure 21 demonstrates the number of alive nodes by applying the contestant algorithms. As shown in this figure, TEDG keeps much more nodes alive compared to others over time. The proposed algorithm brings about 243 and 483 alive nodes in 250 and 500-node WSNs after 1500 rounds. TPSO-CR, DAMS, TEFCSRP, and CAT-EDP keep 241, 232, 233, and 232 nodes alive in 250-node WSNs after 1500 rounds of execution. These values are equal to 450, 274, 479, and 325 in 500-node WSNs. The higher lifetime and number of alive nodes of TEDG is due to that it has a low standard deviation of the exhausted energy by nodes. Accordingly, the nodes consume energy evenly throughout the network and die later. Considering the results of Figs. 17 and 21, we can conclude that the dead nodes are the adopted ones as CH. TPSO-CR also has a low death rate. However, as illustrated in Fig. 15, its packet loss is high.

6.5 Time complexity comparison

The time complexity of an algorithm indicates its usability in real-world scenarios. Figure 22 compares the time complexity of the contestant algorithm versus different numbers of nodes. The average required time to execute TEDG, TPSO-CR, DAMS, TEFCSRP, and CAT-EDP is equal to 156, 151, 37, 121, and 14 s, respectively. It should be mentioned that the considered algorithms are centralized and hence, the reported results present the required time to execute them in the sink or another server. The reason for the low time complexity of DAMS and CAT-EDP is that they are greedy algorithms. TEFCSRP used fuzzy and meta-heuristic algorithms for clustering. Hence, the time complexity of this algorithm becomes higher than that of DAMS. The high time complexity of TPSO-CR is due to applying meta-heuristic algorithms for clustering and tree construction. TEDG improves the convergence speed of the tree construction scheme compared to TPSO-CR. The algorithm has an additional watchdog selection phase compared to TPSO-CR. Therefore, its time complexity does not differ from that of TPSO-CR noticeably.

6.6 Overhead comparison

The imposed overhead by an algorithm indicates its usability in real-world scenarios. The overhead of the considered trust-aware data gathering algorithms is quantified as the exhausted energy by the SFA mitigation mechanism. It includes the consumed energy by watchdogs, CHs, or other nodes to monitor their neighbors. Figure 23 compares the overhead of the contestant algorithms versus different numbers of nodes. TPSO-CR did not provide any SFA mitigation mechanism and hence, it is not included in this figure. The reported results indicate that the proposed algorithm has a lower overhead compared to the competitive algorithm. More precisely, its consumed energy for monitoring tasks is 28%, 64%, 71%, less than that of DAMS, TEFCSRP, and CAT-EDP, respectively. The low overhead of TEDG is due to that it minimizes the number of watchdogs as much as possible. Additionally, watchdogs only monitor CHs and relay nodes. On the other hand, the watchdogs per cluster are independently adopted in DAMS. Therefore, the number of watchdogs and their exhausted energy of this algorithm is more than that of TEDG. In the TEFCSRP algorithm, each sensor monitors its close neighbors. Additionally, in CAT-EDP, each node is monitored by all nodes in its cluster. These two algorithms did not specify watchdogs and used all nodes for trust computation. Accordingly, their monitoring overhead is considerable.

7 Conclusion

This paper proposed TEDG to improve trustworthiness and energy consumption in WSNs. The proposed scheme comprised three phases of clustering, tree construction, and watchdog selection, which were solved using TVAC-PSO. The considered criteria in these phases are the trust and energy level of nodes. Most of the existing research did not include all the mentioned phases, and some studies proposed greedy approaches to solve the phases. Furthermore, most of the algorithms, which used meta-heuristic schemes, proposed ineffective particle representation and objective functions that yielded poor solutions. As confirmed by the experimental results, TEDG outperformed the existing approaches in terms of packet loss and energy-related measures.

As future work, we plan to apply other meta-heuristics, such as Grey Wolf Optimizer (GWO) and Gravitational Search Algorithm (GSA) to solve the intended problem. These algorithms outperform PSO (Rashedi et al. 2009; Mirjalili et al. 2014) and hence, using them for data gathering in WSNs would yield better results. It is also possible to combine some meta-heuristics to improve results. The other direction is to consider more realistic settings. Examples of real-world assumptions are the unreliability of wireless links, packet collision, and node failure. We also aim to include a congestion control mechanism in our design, which is caused by different factors such as packet collision and node buffer overflow (Ghaffari 2015).

Author contributions Conceptualization was contributed by LF; methodology was contributed by KS and LF; software was

contributed by KS; supervision was contributed by LF; validation was contributed by LF; writing—original draft was contributed by LF and MAB; visualization was contributed by KS, writing—review and editing, was contributed by KS, LF, and MAB.

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Competing interests The authors have no relevant financial or non-financial interests to disclose.

References

Abdellatif T, Mosbah M (2020) Efficient monitoring for intrusion detection in wireless sensor networks. Concurr Comput Pract Exp 32:1–13. https://doi.org/10.1002/cpe.4907

Ansari MD, Gunjan VK, Rashid E (2021). On security and data integrity framework for cloud computing using tamper-proofing. In: Kumar, A., Mozar, S (eds) ICCCE 2020. Lecture notes in electrical engineering, vol 698. Springer, Singapore. https://doi.org/10.1007/978-981-15-7961-5_129

Arora VK, Sharma V, Sachdeva M (2020) A multiple pheromone ant colony optimization scheme for energy-efficient wireless sensor networks. Soft Comput 24:543–553. https://doi.org/10.1007/s00500-019-03933-4

Bangotra DK, Singh Y, Selwal A et al (2021) A trust based secure intelligent opportunistic routing protocol for wireless sensor networks. Wirel Pers Commun. https://doi.org/10.1007/s11277-021-08564-3

Bouali T, Senouci S-M, Sedjelmaci H (2016) A distributed detection and prevention scheme from malicious nodes in vehicular networks. Int J Commun Syst 29:1683–1704. https://doi.org/10.1002/dac.3106

Cai RJ, Li XJ, Chong PHJ (2019) An evolutionary self-cooperative trust scheme against routing disruptions in MANETs. IEEE Trans Mob Comput 18:42–55. https://doi.org/10.1109/TMC.2018.2828814

Edla DR, Kongara MC, Cheruku R (2019) A PSO based routing with novel fitness function for improving lifetime of WSNs. Wirel Pers Commun 104:73–89. https://doi.org/10.1007/s11277-018-6009-6

Elhabyan RSY, Yagoub MCE (2015) Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network. J Netw Comput Appl 52:116–128. https://doi.org/10.1016/j.jnca.2015.02.004

Fang J, Feng J (2018) Using PSO-TVAC to improve the performance of DV-Hop. Int J Wirel Mob Comput 14:358–361. https://doi.org/10.1504/IJWMC.2018.10015092

Fang W, Zhang W, Yang W et al (2021) Trust management-based and energy efficient hierarchical routing protocol in wireless sensor networks. Digit Commun Netw 7:470–478. https://doi.org/10.1016/j.dcan.2021.03.005

Ghaffari A (2015) Congestion control mechanisms in wireless sensor networks: a survey. J Netw Comput Appl 52:101–115. https://doi.org/10.1016/j.jnca.2015.03.002

Gunjan VK, Kumar A, Rao AA (2015) Present and future paradigms of cyber crime and security majors—growth and rising trends. In: Proceedings—2014 4th international conference on artificial intelligence with applications in engineering and technology, ICAIET 2014. pp 89–94. https://doi.org/10.1109/ICAIET.2014.24

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

Heinzelman WB, Chandrakasan AP, Balakrishnan H (2002) An application-specific protocol architecture for wireless microsensor networks. Trans Wirel Commun 1:660–670. https://doi.org/10.1109/TWC.2002.804190

Hou J, Qiao J, Han X (2022) Energy-saving clustering routing protocol for wireless sensor networks. IEEE Sens J 22:2845–2857. https://doi.org/10.1109/JSEN.2021.3132682

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

Isaac Sajan R, Jasper J (2021) A secure routing scheme to mitigate attack in wireless adhoc sensor network. Comput Secur 103:1–14. https://doi.org/10.1016/j.cose.2021.102197

Khalid NA, Bai Q, Al-Anbuky A (2019) Adaptive trust-based routing protocol for large scale WSNs. IEEE Access 7:143539–143549. https://doi.org/10.1109/ACCESS.2019.2944648

Li Y, Berjab N, Le Hanh H, Yokota H (2019) Centralized trust scheme for cluster routing of wireless sensor networks. IEEE Int Conf Big Data Big Data 2019:5239–5248. https://doi.org/10.1109/BigData47090.2019.9006214

Mann PS, Singh S (2019) Improved metaheuristic-based energy-efficient clustering protocol with optimal base station location in wireless sensor networks. Soft Comput 23:1021–1037. https://doi.org/10.1007/s00500-017-2815-0

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey Wolf Optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mittal N, Singh U, Sohi BS (2019) An energy-aware cluster-based stable protocol for wireless sensor networks. Neural Comput Appl 31:7269–7286. https://doi.org/10.1007/s00521-018-3542-x

Mittal N, Singh S, Singh U, Salgotra R (2021) Trust-aware energy-efficient stable clustering approach using fuzzy type-2 Cuckoo search optimization algorithm for wireless sensor networks. Wirel Netw 27:151–174. https://doi.org/10.1007/s11276-020-02438-5

Monnet Q, Mokdad L, Ballarini P et al (2017) DoS detection in WSNs: Energy-efficient methods for selecting monitoring nodes. Concurr Comput Pract Exp 29:e4266. https://doi.org/10.1002/cpe.4266

Ni Q, Pan Q, Du H et al (2017) A novel cluster head selection algorithm based on fuzzy clustering and particle swarm optimization. IEEE/ACM Trans Comput Biol Bioinforma 14:76–84. https://doi.org/10.1109/TCBB.2015.244647

Pachlor R, Shrimankar D (2018) LAR-CH: a cluster-head rotation approach for sensor networks. IEEE Sens J 18:9821–9828. https://doi.org/10.1109/JSEN.2018.2872065

Pavani M, Rao PT (2019) Adaptive PSO with optimised firefly algorithms for secure cluster-based routing in wireless sensor networks. IET Wirel Sens Syst 9:274–283. https://doi.org/10.1049/iet-wss.2018.5227

Prabhu S, Mary Anita EA (2020) Trust based secure routing mechanisms for wireless sensor networks: a survey. IEEE Int Conf Adv Comput Commun Syst ICACCS 2020:1003–1009. https://doi.org/10.1109/ICACCS48705.2020.9074464

Prithi S, Sumathi S (2020) LD2FA-PSO: a novel learning dynamic deterministic finite automata with PSO algorithm for secured energy efficient routing in wireless sensor network. Ad Hoc Netw 97:102024. https://doi.org/10.1016/j.adhoc.2019.102024

Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248. https://doi.org/10.1016/j.ins.2009.03.004

Ratnaweera A, Halgamuge SK, Watson HC (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. Trans Evol Comp 8:240–255. https://doi.org/10.1109/TEVC.2004.826071

Rodrigues P, John J (2020) Joint trust: an approach for trust-aware routing in WSN. Wirel Netwo 26:3553–3568. https://doi.org/10.1007/s11276-020-02271-w

Saidi A, Benahmed K, Seddiki N (2020) Secure cluster head election algorithm and misbehavior detection approach based on trust management technique for clustered wireless sensor networks. Ad Hoc Netw 106:102215. https://doi.org/10.1016/j.adhoc.2020.102215

Sajan RI, Christopher VB, Kavitha MJ, Akhila TS (2022) An energy aware secure three-level weighted trust evaluation and grey wolf optimization based routing in wireless ad hoc sensor network. Wirel Netw 28:1439–1455. https://doi.org/10.1007/s11276-022-02917-x

Sánchez-Casado L, Maciá-Fernández G, García-Teodoro P, Magán-Carrión R (2015) A model of data forwarding in MANETs for lightweight detection of malicious packet dropping. Comput Netw 87:44–58. https://doi.org/10.1016/j.comnet.2015.05.012

Shahid J, Muhammad Z, Iqbal Z et al (2022) Cellular automata trust-based energy drainage attack detection and prevention in wireless sensor networks. Comput Commun 191:360–367. https://doi.org/10.1016/j.comcom.2022.05.011

Shcherba EV, Litvinov GA, Shcherba MV (2019) A novel reputation model for trusted path selection in the OLSR routing protocol. IEEE Int Conf Inf Sci Commun Technol Appl Trends Oppor, ICISCT 2019. https://doi.org/10.1109/ICISCT47635.2019.9011870

Shyama M, Pillai AS, Anpalagan A (2022) Self-healing and optimal fault tolerant routing in wireless sensor networks using genetical swarm optimization. Comput Netw 217:109359. https://doi.org/10.1016/j.comnet.2022.109359

Singh MK, Amin SI, Imam SA et al (2018) A survey of wireless sensor network and its types. IEEE Int Conf Adv Comput Commun Control Netw ICACCCN 2018:326–330. https://doi.org/10.1109/ICACCCN.2018.8748710

Supriya M, Adilakshmi T (2022) Secure cluster-based routing using modified spider monkey optimization for wireless sensor networks. In: Bhateja V, Satapathy SC, Travieso-Gonzalez CM, Adilakshmi T (eds) Smart Intelligent Computing and Applications, 1. Smart Innovation, Systems and Technologies, 282. Springer, Singapore. https://doi.org/10.1007/978-981-16-9669-5_23

Swapnarani P, Rao PR, Gunjan VK (2022) Self defence system for women safety with location tracking and SMS alerting through GPS and GSM networks. In: Gunjan VK, Zurada JM (eds) Modern Approaches in Machine Learning & Cognitive Science: A Walkthrough.Studies in Computational Intelligence, vol 1027., 1st edn. Springer, pp 361–368

Tomić I, McCann JA (2017) A survey of potential security issues in existing wireless sensor network protocols. IEEE Internet Things J 4:1910–1923. https://doi.org/10.1109/JIOT.2017.2749883

Wang B, Chen X, Chang W (2014) A light-weight trust-based QoS routing algorithm for ad hoc networks. Pervasive Mob Comput 13:164–180. https://doi.org/10.1016/j.pmcj.2013.06.004

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

Wu J, Song C, Fan C et al (2019) DENPSO: A distance evolution nonlinear PSO algorithm for energy-efficient path planning in 3D UASNs. IEEE Access 7:105514–105530. https://doi.org/10.1109/ACCESS.2019.2932148

Yang H, Zhang X, Cheng F (2021) A novel algorithm for improving malicious node detection effect in wireless sensor networks. Mob Netw Appl 26:1564–1573. https://doi.org/10.1007/s11036-019-01492-4

Yin H, Yang H, Shahmoradi S (2022) EATMR: an energy-aware trust algorithm based the AODV protocol and multi-path routing approach in wireless sensor networks. Telecommun Syst 81:1–19. https://doi.org/10.1007/s11235-022-00915-0

Yun J, Seo S, Chung JM (2018) Centralized trust-based secure routing in wireless networks. IEEE Wirel Commun Lett 7:1066–1069. https://doi.org/10.1109/LWC.2018.2858231

Zhao C, Wu C, Wang X et al (2017) Maximizing lifetime of a wireless sensor network via joint optimizing sink placement and sensor-to-sink routing. Appl Math Model 49:319–337. https://doi.org/10.1016/j.apm.2017.05.001

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