Journal of Distributed Computing and Systems (JDCS)
Vol. 8, Issue 1, 2025, Pages 8-25

A Trust-based Clustering Algorithm in Homogeneous Wireless Sensor Networks

Hossein Jadidoleslamy
Department of Computer Engineering, University of Zabol (UOZ), Zabol, Sistan and Balouchestan, IRAN

Article History:
Received: 10 May 2025
Received in revised form: 25 July 2025
Accepted: 14 August 2025
Available online: 15 September 2025

Abstract

Wireless Sensor Networks (WSNs) exhibit a plethora of applications in the domains of monitoring and tracking. These networks typically comprise numerous sensor nodes in conjunction with a Sink. Challenges and attributes such as complex organization and management, dynamic network topology, elevated node density, excessive redundant data, limited scalability, constrained resources, security vulnerabilities, and errant node behavior substantially impair their overall efficiency. A viable resolution to these challenges is the implementation of trust-based clustering through the utilization of a Trust Management System (TMS). So, this paper proposes a trust-based clustering algorithm for homogeneous WSNs, called TSC-WSN. In TSC-WSN, the trustworthiness of the nodes constitutes the primary criterion in the clustering procedure; it enhances the collaboration among the nodes and fortifies the security of the Wireless Sensor Network (WSN). TSC-WSN is distinct from other clustering schemes concerning the criteria and procedures utilized for calculating and predicting the trustworthiness of nodes, electing the cluster heads, and forming the clusters. Finally, the performance of TSC-WSN is compared to the performance of Zhao et al. and Zeng et al. clustering algorithms; the results derived from simulations and statistical analysis indicate that TSC-WSN is improved in the domains of accuracy regarding the identification of untrusted nodes, scalability, fault tolerance, and the mean of packet loss rate.

Keywords: Wireless Sensor Networks (WSNs), Clustering, Trust Management System (TMS), Trust-based Clustering.

I. Introduction

Wireless Sensor Networks (WSNs) typically comprise numerous small sensor nodes and a central Sink. These networks are characterized by their infrastructure-less, data-centric, and application-oriented nature. Various challenges and features, such as the difficulty in organizing and managing large-scale WSNs, dynamic topology, high node density, excessive redundant data, limited scalability, constrained resources, security vulnerabilities, and node misbehavior due to malicious, compromised, or selfishness motives, significantly decrease their overall performance [1, 2, 3, 4].

Clustering is an effective solution for addressing the mentioned problems. In the clustered WSNs, nodes are classified into virtual groups, called clusters. Each cluster consists of a cluster head (CH) and its member nodes. The members of a cluster do not communicate directly with the Sink; instead, they send their data to the CH .The CH serves as an intermediary between the cluster's members and the Sink; it collects data from the cluster's members, aggregates them, and subsequently routes and forwards them to the Sink. CHs form the communication backbone for data transmission within the clustered WSNs. Clustering algorithms typically include three primary phases, including clusters' formation, CHs' election, and maintenance [5, 6, 7]. In continuation, Figure 1 illustrates the different aspects for using the clustering algorithms in WSNs. However, most of clustering algorithms have weaknesses such as low security, high traffic and computational overhead, significant resource consumption, low accuracy, and incompatibility with the especial characteristics of WSNs [8, 9, 10], like severe resource constraints, dynamic network topology, and nodes' mobility. Therefore, efficient clustering is a vital necessity for WSNs.

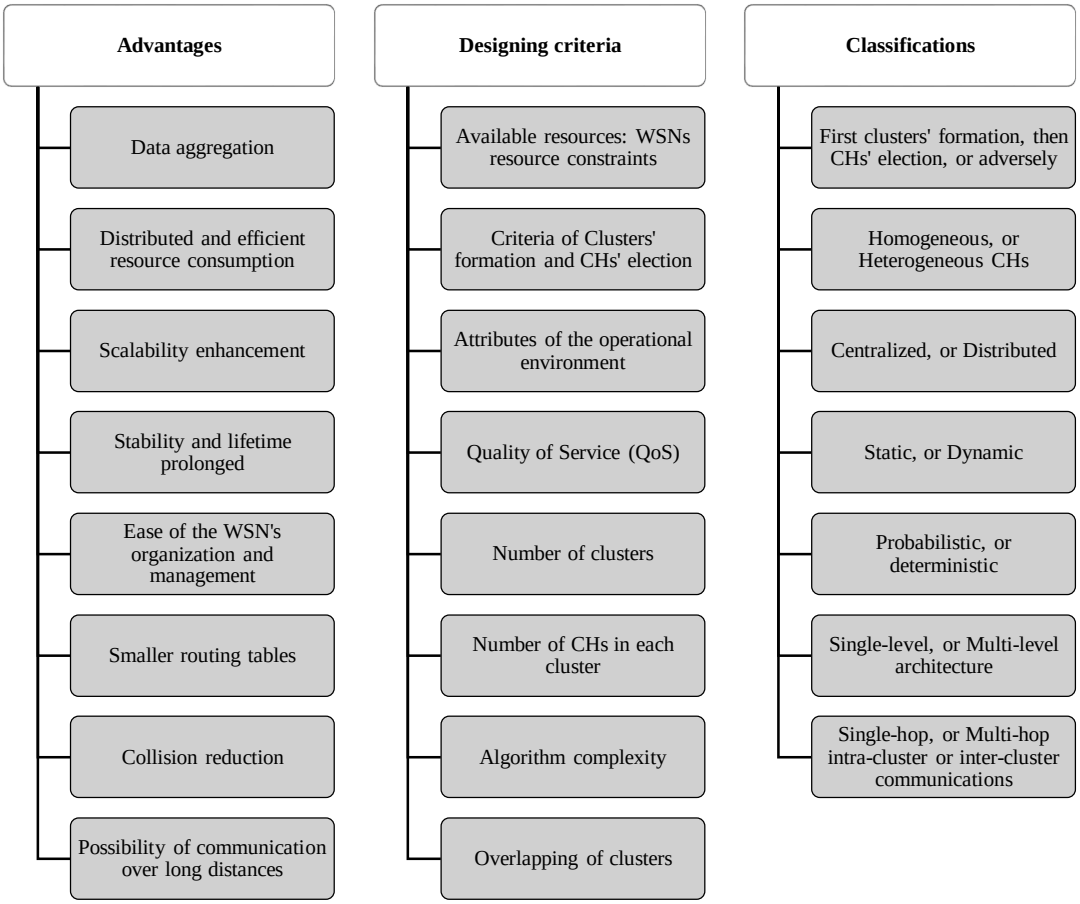


Figure 1. Different aspects for using clustering algorithms in WSNs [5-19]

Another effective solution to the aforementioned problems is the Trust Management System (TMS) [6, 7, 20, 21, 22]; Figure 2 illustrates the different aspects for using the Trust Management Systems (TMSs) in WSNs [6-9]. The trust concept is inspired by the human interactions in the human societies; it is a context-dependent, dynamic, and complex concept in the WSNs. TMSs evaluate the trustworthiness of nodes, identify the untrustworthy nodes (malicious, compromised, or selfishness), and eliminate them from the network's operations. They enhance cooperation among the sensor nodes, and improve the network security and its performance [20, 21, 22]. However, current TMSs possess weaknesses, including high computational complexity, high traffic and processing overhead, significant resource consumption, low accuracy, a high error rate in detecting untrustworthy nodes, and incompatible with the especial characteristics of WSNs like severe resource constraints, high density of sensor nodes, dynamic network topology, and sensor nodes' mobility. Therefore, it is essential to design an adaptive TMS suitable for these networks.

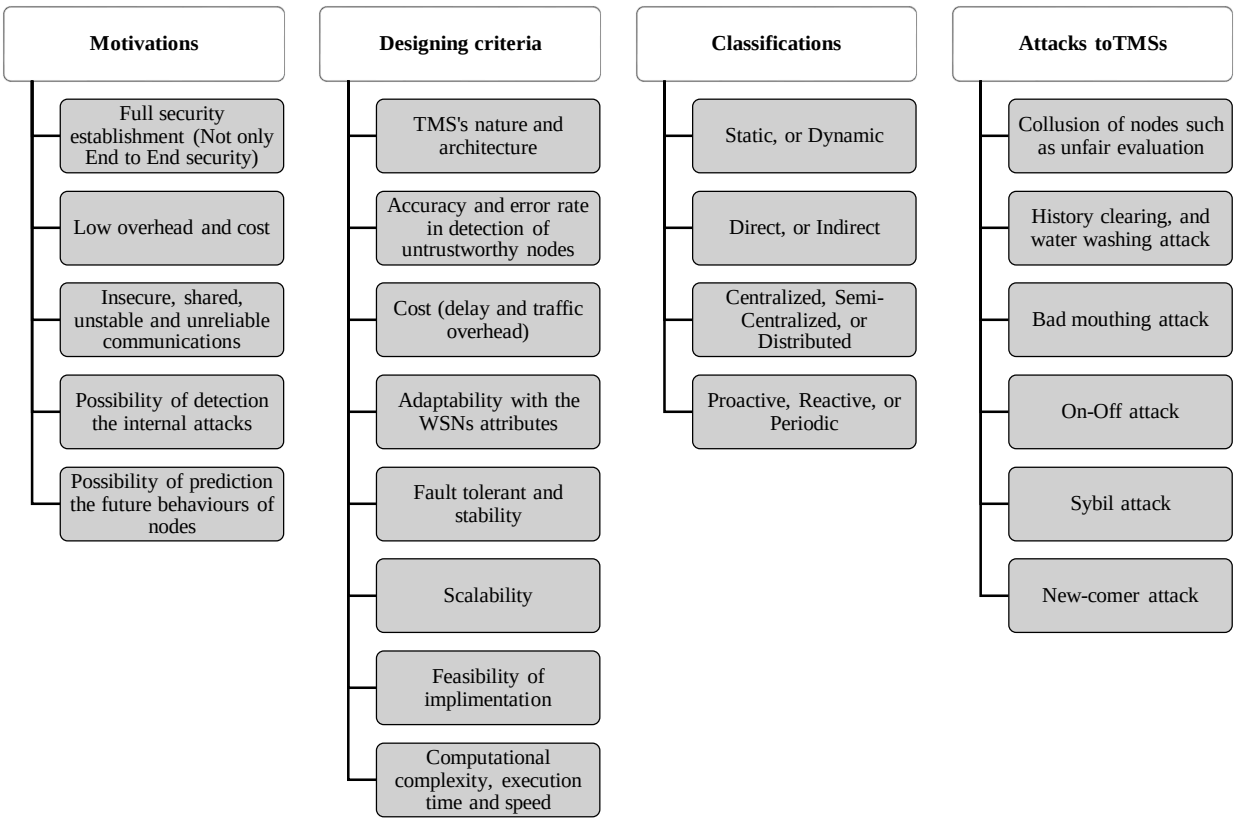


Figure 2. Different aspects for using TMSs in WSNs: motivations, key factors to consider in the designing step, different categories of TMSs, and potential attacks targeting TMSs [6, 7, 8, 9]

As a result, this paper proposes a trust-based clustering algorithm for homogeneous WSNs, called TSC-WSN. This algorithm considers the trustworthiness of nodes as the main metric in the clustering process, which includes the trust-based CHs' election and clusters' formation; it also identifies the untrustworthy nodes, and eliminates them from the network's operations. TSC-WSN is distinct from other trust-aware clustering schemes concerning the criteria and procedures utilized for calculating and predicting the trustworthiness of nodes, electing the cluster heads, and forming the clusters. In continuation, Figure 3 illustrates the key steps involved in the TSC-WSN.

Finally, the performance of the TSC-WSN is compared to the performance of the clustering algorithms proposed by Zhao et al. [33] and Zeng et al. [34]; the results of simulations utilizing TRMSim-WSN and NS2 simulators, and statistical analysis utilizing Expert Choice and Grey Relationship Analysis tools, demonstrate that the TSC-WSN is improved in various parameters like accuracy in the detection of untrusted nodes, scalability, fault tolerance, and the average of packets loss rate. Nevertheless, its performance is decreased regarding resource consumption (including energy, bandwidth, and memory), traffic and processing overhead, computational complexities, and execution speed.

The subsequent sections of this paper are structured as follows: Section 2 expresses the related works; it reviews some of the popular clustering algorithms, in summary; Section 3 describes the different steps of the proposed trust-based clustering algorithm for homogeneous WSNs (i.e., TSC-WSN), in detail; Section 4 evaluates the performance of TSC-WSN and compares it to the Zhao et al. and Zeng et al. Clustering algorithms; it briefly explains the results of simulations and statistical analysis; and finally, Section 5 provides a synthesis of the paper's findings while outlining prospective avenues for the future investigation.

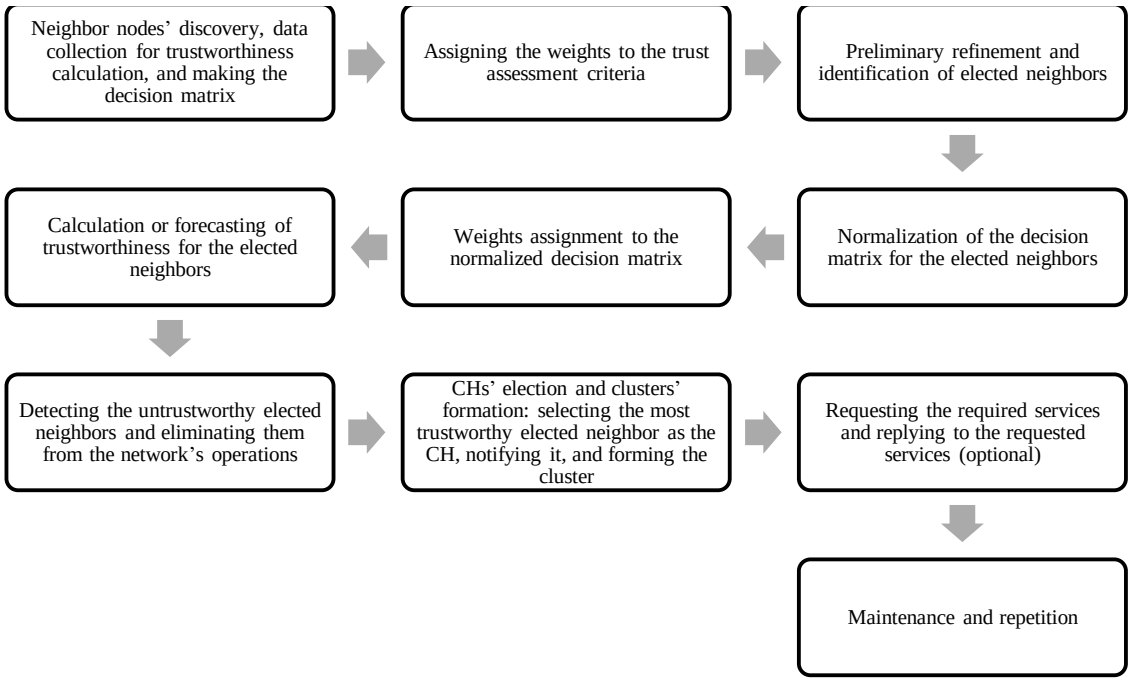


Figure 3. The main steps of the TSC-WSN

II. Related Works

This section provides a comprehensive review of various clustering algorithms for different networks such as Mobile Adhoc Networks (MANETs) and WSNs.

Park et al. [23], designed a trust-aware clustering scheme. In this scheme, each node evaluates the trust value of its neighbors and recommends the one with the highest trust value as its trust guarantor. Then the recommender node becomes a member of CH which is one-hop away. When nodes recommend a CH, they give a recommendation certificate (R-Certificate) to the CH which is used to authenticate. So, the CH which has many R-Certificates, is more trusted and the new cluster in the new place refers to the node trust value by the previous CH for the trust evaluation. Crosby et al. [24], proposed a distributed trust-aware framework and a mechanism for the election of trustworthy CHs. The proposed mechanism reduces the likelihood of compromised or malicious nodes from being selected as CHs. They proposed a trust-aware CH election scheme where each node gives a trust value to other nodes according to their behavior and highly trusted nodes become CHs. A node's behavior is judged by counting the frequency of its successful and unsuccessful transmissions. When a new CH should be elected, some nodes with a high reputation value are recommended for the CH role by the members, and one of them is selected as a new CH by the current CH. Kadari et al. [25], proposed a secured weight-aware clustering algorithm; this algorithm elects the CHs according to their weight computed by combining a set of parameters such as stability, battery, and degree. To create or maintain a clustering architecture, first, the Discovery stage should be done, during which information about the neighborhood should be retrieved. For this purpose, nodes desiring to be CH send CH_ready beacons to their D hops radius. Then, nodes receiving this beacon, estimate a trust value and send it back. After a discovery period, nodes having initiated this operation can derive from the received responses such as degree, stability, and trust value. Then each node adds to the previous parameters the state of its battery and the maximum value and combines them to compute the global weight. After nodes choose the CH that has the maximum weight, each newly elected CH needs to discover each other to construct a virtual backbone for inter-cluster communication. Thus, every newly elected CH broadcasts a discovery request over the network. CHs receiving this request register the certificate of the new CH and send their certificate. Peng et al. [26], proposed a voting-aware clustering algorithm; it is a trust-aware clustering scheme that evaluates the stability of nodes through computing the neighbor change ratio and the residual power of nodes. In this scheme, each node votes for other nodes only if the node is the most trustworthy among its neighbors. Votes are propagated only to one-hop neighbors, and they are not forwarded by other nodes. For clustering, these steps are followed: calculating the stability of each node, computing the trust of each node to its neighbors, and finally, each node voting its neighbors according to the voting algorithm. Now, it chooses the large vote as the CH; but if the number of votes is the same, it chooses the best stability as the CH. If the number of votes and stability are the same, it chooses the smallest ID as CH. Song et al. [27], proposed the Trust-aware Low Energy Adaptive Clustering Hierarchy Protocol (T-LEACH) for WSNs. It contains two main components: the Monitoring module

and the Trust Evaluation module. Each node also maintains a Neighbor Situational Trust Table filled with trust value entries for each pair of node IDs and situational operations. The trust update slot allows the CH to share its trust values with its cluster's members. T-LEACH loses fewer data than the LEACH algorithm because half of all data sent by cluster members is received by the gateway. However, T-LEACH is unable to stop the constant loss of data, because of the lack of monitoring on the CH. Ferdous et al. [28], presented a CH election algorithm that includes a trust interaction table that holds the details of nodes' maximum neighbors. The node which includes the maximum number of neighbors is elected as CH and others are made as cluster members; i.e., after deployment, the nodes broadcast their ID and trust value to their neighbors along with the REQ/REPLY flag. When the participating nodes have discovered their neighbors, they exchange information about the number of one-hop neighbors. The node that has maximum neighbors from the trust interaction table is selected as CH and other nodes become members of cluster or local nodes. The CH(s) election is distributed and secured. To formalize the trust value of a particular node, nodes monitor its behavior to collect information from its neighbors and then decide on the node. They have used a quantitative trust evaluation algorithm at each node to evaluate the direct trust of its neighbor nodes. Holczer et al. [29], proposed a completely hidden election scheme; they designed an anonymous aggregator election and data aggregation scheme for WSNs. This scheme consists of two steps: first, each member elects himself as a CH according to the probability that a member becomes a CH; then, each member checks if there is a member who elected himself as a CH. The check reveals only the existence or not the CH node, and none of the members knows which node is the CH in the cluster. Nishimura et al. [30], proposed a scheme where all nodes give a trust value to each elected, and the most trusted nodes become CH. Otherwise, nodes join a nearby cluster to form clusters in the network. This scheme requires a lot of communication overhead to build a trust evaluation system. Moreover, this scheme burdens a few CH nodes with a lot of normal nodes for a long time. Wang et al. [31], presented a secure clustering scheme that divides the WSN into several clusters and applies the mesh topology structure. The CH is selected within the cluster according to the number of trusted connections, and the nodes that have trusted connections with the CH will be the core nodes. At first, the cluster's nodes set their trust values to 0. The cluster service group is made up of CH and core nodes, which can join together to form the service group, that is in charge of providing service for various requests from the cluster's members. The nodes that connect to the service group will be peripheral nodes and do nothing except forward the received messages. The messages between different clusters will be forwarded by the CHs and due to the existence of the session keys between the CHs, the messages can be transmitted in the common channel. Sheik Dawood et al. [32], presented an efficient Clustering architecture for HWSN. Clustering is one of main technique to condense the energy consumption in the network. Selecting Cluster head is the major process for energy efficiency of clustering algorithms. As maximum energy is dissipated during data transfer, communication within the cluster is paramount. Communication distance between the Cluster head and member node is paramount. Node with elevated communication distance within the Cluster will acquire more energy. So the proposed protocol reduces the communication distance between in the cluster to condense the energy burning up in the network and recover the lifetime of the network. The main purpose of the proposed algorithm is to reduce the energy efficiency in terms of intra cluster communication. The distance is most important in the wireless communication where the longer distance need more power dissipation than the shorter distance. So among the distributed nodes deployed in the field, CH location among the nodes makes the energy efficient communication between the nodes. Zhao et al. [33], is studied and aimed to solve the problems of unreasonable cluster-head selection and excessive energy consumption in LEACH. In order to overcome drawbacks of unreasonable cluster-head selection and excessive energy consumption in wireless sensor networks (WSNs), a modified cluster-head selection algorithm based on LEACH (LEACH-M) was proposed. Based on distributed address assignment mechanism (DAAM) of ZigBee, both residual energy and network address of nodes were taken into account to optimize cluster-head threshold equation. Furthermore, by leveraging a cluster-head competitive mechanism, LEACH-M successfully balanced the network energy burden and dramatically improved energy efficiency. Zeng et al. [34], proposes a sector clustering algorithm based on K-means, called KMSC. KMSC improves efficiency and balances the cluster size by employing symmetric dividing sectors in conjunction with K-means. For the selection of cluster heads (CHs), KMSC uses the residual energy and distance to calculate the weight of the node, then selects the node with the highest weight as CH. A hybrid single-hop and multi-hop communication is utilized to reduce long-distance transmissions. Furthermore, the impact of the number of sectors, the threshold for clustering, and the network size on the performance of KMSC has been explored. Hu et al. [35], introduces QPSOFL, a clustering and routing protocol that integrates quantum particle swarm optimization and a fuzzy logic system to enhance energy efficiency and prolong network lifespan. QPSOFL employs an enhanced quantum particle swarm optimization algorithm to select optimal cluster heads, utilizing Sobol sequences for population diversification during initialization. Additionally, it incorporates Lévy flight and Gaussian perturbation-based position updates to prevent trapping in local optima. Benchmark experiments validate QPSOFL's efficacy compared to Harris Hawks Optimization (HHO), Grey Wolf Optimization (GWO),

Particle Swarm Optimization (PSO), and Quantum Particle Swarm Optimization (QPSO), focusing on accuracy, search capability, and convergence speed. Within QPSOFL, a fuzzy logic system determines the best next-hop cluster head based on descriptors such as residual energy, energy deviation, and relay distance. Rajalingam et al. [36], offers a reinforcement learning (RL) based energy-aware clustering approach, whereby peripheral cluster nodes monitor environmental factors like energy use and choose an optimal cluster leader (CH). Connect the CH (BS) to the base station. In the simulation (PDR), performance factors such as network lifetime, energy tax, network stability period, and packet delivery rate are all taken into account. Al-Sulaifanie et al. [37], investigates many drawbacks and limitations of current clustering algorithms designed for wireless sensor networks. These limitations include high resource requirements, significant overhead, tight synchronization requirements, and TDMA limitations. A new clustering algorithm called hybrid access and an adaptive duty cycle clustering (HADC) protocol is introduced, aiming to address the previous limitations. HADC protocol is based on several concepts, including static clusters, adaptive duty cycling, hybrid scheme channel access, minimum node's functionality, and pseudo synchronization. The resulting network's important features are energy efficiency, robustness, low latency, scalability, and self-healing. Swarm intelligence (SI)-based metaheuristics are well applied to solve real-time optimization problems of efficient node clustering and energy-aware data routing in wireless sensor networks. Mann et al. [38], presents another superior approach for these optimization problems based on an artificial bee colony metaheuristic. The proposed clustering algorithm presents an efficient cluster formation mechanism with improved cluster head selection criteria based on a multi-objective fitness function, whereas the routing algorithm is devised to consume minimum energy with least hop-count for data transmission. Chauhan et al. [39], proposed an energy-aware unequal clustering algorithm (EAUCA) to diminish the energy holes and enhance the network's lifetime. The proposed EAUCA creates the unequal sized clusters in such a way that clusters in the base station vicinity are smaller than the farthest. The competition radius to divide the network into unequal clusters is decided by considering the node's residual energy and distance to the base station. The cluster heads are selected based on remaining energy and a node's degree in a competition radius. In the inter-cluster data forwarding, the role of relay node and cluster head is also detached, which reduces the data traffic load of the cluster head nodes. The nodes with low node's degree relay the inter-cluster traffic to the base station.

III. TSC-WSN: A Trust-based Clustering Algorithm for Homogeneous Wireless Sensor Networks

This section describes the different steps of the proposed trust-based clustering algorithm tailored for homogeneous WSNs, called TSC-WSN, in detail. In TSC-WSN, each node evaluates the trustworthiness of its neighboring nodes through the execution of multiple transactions and the observation of their behaviors; i.e., it quantifies their trustworthiness based on its empirical observations and measurements. Also, each node can predict the trustworthiness of its neighboring nodes according to the previous trust values of them. Then, it detects untrustworthy neighbors and establishes trust relationships by eliminating them from the network's operations. Subsequently, it designates the most trustworthy neighbor as the CH and communicates this decision accordingly. After that, the selected CH determines its cluster's members. Finally, the cluster's members request the required services, and the CH provides the requested services. The TSC-WSN enhances the network security and improves its performance by excluding untrustworthy nodes from the network's operations and refraining from designating them as members of clusters or as CHs. The TSC-WSN segments time into discrete time intervals. During the first time interval, each node assigns the primary trust values to its neighboring nodes. Trust is represented as a numerical value ranging from 0 to 1; it assumes that the initial trust value for each neighbor is established at 0.5. Then, each node calculates the trustworthiness of its neighboring nodes by monitoring their behaviors and collecting the requisite information. The proposed criteria for the calculation of trustworthiness within the context of trust-based clustering are presented in Table 1.

Table 1. The proposed criteria for the calculation of trustworthiness

No.	Criterion
1	Available resources, especially residual energy and empty buffer
2	Congestion status
3	Accuracy and reliability of the received data
4	Distance to the Sink
5	Distance to the gravity and geometric center of the cluster
6	Stability during a time interval
7	Density around the node and the number of its neighbors
8	The number of times it has already been CH
9	The recommendations of security systems about it
10	Its previous trust values

Since trust is a context-dependent concept, the criteria for evaluating the trustworthiness of nodes are depend on the level of accuracy and security required, possible attacks, intended application and the operating environment, the resources of the WSN and its nodes' capabilities [7, 8]. As the number of monitored criteria

and behaviors increases, the accuracy, cost, and security level will be more, but the feasibility of its implementation will be less. In continuation, the various steps of the TSC-WSN are explained.

A. *Neighbor nodes' discovery, data collection for tustworthiness calculation, and making the decision matrix*

At the outset, each node identifies its neighboring nodes, and systematically observes the various events, activities, and corresponding behaviors of them; subsequently it determines their specifications utilizing the overhearing, eavesdropping, or passive information-gathering techniques; then, it analyzes and stores the collected data within the decision matrix (D), which is characterized by dimensions (n×m), defined as follows:

$$D = \begin{bmatrix} X_{11} & \cdots & X_{1m} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{nm} \end{bmatrix}$$

n: represents the number of the neighboring nodes associated with the node (the rows of the decision matrix); i= {1, … , n}

m: denotes the number of criteria utilized for trust assessment (the columns of the decision matrix); j= {1, … , m}

X_{ij}: signifies the evaluation score assigned to the i'th neighboring node in relation to the j'th criteria for trust assessment

B. *Assigning the weights to the trust assessment criteria*

Given that the aforementioned trust assessment criteria possess varying degrees of significance in the decision-making process, this step assigns their respective weights as follows:

- Normalizing the decision matrix (D) by using the Linear normalization method as follows:

$$P_{ij} = \frac{x_{ij}}{\sum_{i=1}^{n} x_{ij}} \qquad (1)$$

- Calculating the uncertainty factor (E_j), the deviation degree than the acquired information (d_j), and the weights (W_j) attributed to the trust assessment criteria as follows:

$$E_j = \frac{1}{\text{Ln } n} \sum_{i=1}^{n} P_{ij} \text{ Ln } P_{ij},\ E_j \in [0, 1] \qquad (2)$$

$$d_j = 1 - E_j \qquad (3)$$

$$W_j = \frac{d_j}{\sum_{j=1}^{m} d_j} \qquad (4)$$

Consequently, the weights corresponding to the trust assessment criteria are as follows:

$W = \{W_1, W_2, \ldots, W_m\}$, so that: $\sum_{j=1}^{m} W_j = 1,\ 0 \le W_j \le 1$

C. *Preliminary refinement and identification of elected neighbors*

Prior to the calculating or predicting the trustworthiness for all neighboring nodes, some series of initial refinements are conducted to determine the elected neighbors the associated node; this approach reduces the volume of required computational and processing overhead associated with trustworthiness calculation or forecast. So, this step determines certain neighbors of the node as its elected neighbors by defining the specific thresholds. For instance, if the level of available resources, such as residual energy, falls below the designated threshold, or the previous trust values of the neighboring nodes be less than the predefined threshold, those neighbors will be excluded from the list of neighbors; then, calculating or forecasting the trustworthiness will be conducted only for the remaining neighbors of the node.

D. *Normalization of the decision matrix for the elected neighbors*

This step normalizes the decision matrix (D) by mapping its values within the range [0, 1] for the elected neighbors of the associated node; it leads to a normalized and scale-less decision matrix (R). To achieve this, it uses the following normalization manner:

$$P_{ij} = \frac{x_{ij}}{x_i^{max}}, \text{ for positive criterion} \qquad (5)$$

$$P_{ij} = \frac{x_i^{min}}{x_{ij}}, \text{ for negative criterion} \qquad (6)$$

$$R = \begin{bmatrix} P_{11} & \cdots & P_{1m} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{nm} \end{bmatrix}$$

E. *Weights assignment to the normalized decision matrix*

This step involves the allocation of the computed weights (W) to the normalized and scale-less decision matrix (R) through the multiplication of each column by the weight associated with its respective criterion (W_j); subsequently, the weighted normalized and scale-less decision matrixit (V) is constructed as follows:

$$V = W \times R = \begin{bmatrix} V_{11} & \cdots & V_{1m} \\ \vdots & \ddots & \vdots \\ V_{n1} & \cdots & V_{nm} \end{bmatrix} \qquad (7)$$

F. *Calculation or forecasting of trustworthiness for the elected neighbors*

The TSC-WSN calculates or forecasts the trust values of the nodes. For this purpose, each node stores a data set of the previous trust values for its elected neighbors in a table, called DSPT. In this scenario, the node establishes a threshold for the Variations Rate (VR_{thr}) of the previous

trust values of its neighbors; if their VR exceed the predefined threshold ($VR > VR_{thr}$), the node calculates their trust values; otherwise, it forecasts their trust values over one or more time intervals utilizing the DSPT. Some of the commonly utilized indexes for VR calculation are outlined in Table 2.

Table 2. The popular indexes for the calculation of variations rate (VR) [7-9]

No.	VR Index	Formula
1	Variations Range (VRange)	VRange = (Maximum value in the DSPT) – (Minimum value in the DSPT)
2	Variance (Var)	$Var = \frac{(T_1 - \overline{T})^2 + (T_2 - \overline{T})^2 + \ldots + (T_n - \overline{T})^2}{n} = \frac{1}{n} \times [T_1{}^2 + T_2{}^2 + \ldots + T_n{}^2] - \overline{T}$ n: size of the DSPT; $\overline{T}$: mean trust value
3	Standard Deviation (SD)	$SD = \sqrt{Var}$
4	Coefficient of Variations (CV)	$CV = \frac{SD}{\text{Average trust value}} = \frac{\sqrt{Var}}{\overline{T}}$

1. Trustworthiness calculation

This step delineates the proposed method for the computation of trust values assigned to nodes. To achieve this objective, each node evaluates the trust values of its elected neighbors by using the weighted normalized and scale-less decision matrix (V) in the following manner:

- Determining of both the positive ideal solution and the negative ideal solution corresponding to each trust calculation criterion, as follows:
  - The positive ideal solution is defined as the optimal value (the best value) for the j'th trust calculation criterion: f_j^*
  - The negative ideal solution is designated as the least favorable value (the worst value) for the j'th trust calculation criterion: f_j^-
- Performing the computation of the value of earning (S) and the value of regret (R) for the elected neighbors of the node in the subsequent manner:
  - Calculation of the relative distance of the i'th elected neighbor from the positive ideal solution (S_i), as follows:

$$S_i = \sum_{j=1}^{m} W_j \times \frac{(f_j^* - f_{ij})}{(f_j^* - f_j^-)} \qquad (8)$$

  - Calculation of the maximum regret of the i'th elected neighbor in relation to the farness or proximity of the positive ideal solution (R_i), as follows:

$$R_i = \text{Max}\ \{W_j \times \frac{(f_j^* - f_{ij})}{(f_j^* - f_j^-)}\} \qquad (9)$$

- The determination of the trustworthiness of each elected neighbor of the node (T) is conducted as follows:

$$T_i = \text{New Trust Value} = \xi \times \frac{(S_i - S^*)}{(S^- - S^*)} + (1 - \xi) \times \frac{(R_i - R^*)}{(R^- - R^*)} \qquad (10)$$

$$S^- = \text{Max}\ \{S_i\},\ S^* = \text{Min}\ \{S_i\} \qquad (11)$$

$$R^- = \text{Max}\ \{R_i\},\ R^* = \text{Min}\ \{R_i\} \qquad (12)$$

$\xi \in [0, 1]$; usually, it is equal to: $\xi = 0.5$

- Subsequently, the node establishes the weights for the new trust values (W_{NT}) and the previous trust values (1- W_{NT}); then, it computes the total trust values (TT) for its elected neighbors by integrating their new and previous trust values, as follows:

$$TT = W_{NT} \times (\text{New Trust Value}) + (1 - W_{NT}) \times (\text{Previous Trust Value}) \qquad (13)$$

Finally, the node stores the computed total trust values for its elected neighbors within its Neighbors' Trust Table (NTT).

2. Trustworthiness forecast

In this step, the TSC-WSN proposes a method for the prediction of the new trust values pertaining to nodes based on their previous trust values. In this case, it is unnecessary to gather information, exchange trust data, or perform trust value calculations. For this purpose:

- each node maintains a data set of the previous trust values (DSPT) for its elected neighbors; subsequently, it predicts their new trust values utilizing the DSPT, alongside a statistical-mathematical quantitative forecasting technique, as follows:

$$T_{n,t} = \frac{\sum_{i=1}^{k} T_{p,t-i}}{k} \qquad (14)$$

k: the number of discrete time intervals utilized for computing the average of the previous trust values

$T_{n,t}$: the new forecasted trust value corresponding to the t'th time interval

$T_{p,i}$: the previous trust value associated with the i'th time interval

- In order to determine the optimal value of the parameter k, it computes the parameter e across various time intervals; the parameter e indicates the forecasting error and it is defined as the difference between the trust values of two previous time intervals. Subsequently, it selects the parameter k that

yields the minimal e value. The average of the forecasting error (e) is computed as follows:

$$e = \frac{\sum_{i=1}^{N} |T_{p,i} - T_{n,i}|}{N} \quad (15)$$

$T_{p,i}$: the actual trust value

$T_{n,i}$: the predicted trust value

- If the trust values of the different time intervals have been different weights, the proposed method integrates them together as follows:

$$T_{n,t} = (W_1 \times T_{p,t-1} + W_2 \times T_{p,t-2} + \ldots + W_k \times T_{p,t-k}) / (\sum_{i=1}^{k} W_i) \quad (16)$$

W_i: the weight assigned to the previous trust value associated with the i'th time interval

It is noteworthy that usually the weights of the newer trust values are more than those assigned to the earlier trust values. Finally, the node stores the predicted trust values for its elected neighbors within its Neighbors' Trust Table (NTT).

G. *Detecting the untrustworthy elected neighbors and eliminating them from the network's operations*

In this step, each node identifies its malicious, compromised, or selfishness (untrustworthy) elected neighbors, based on the trust records housed within its NTT table. To achieve this, it delineates two thresholds of trust values ($Trust_{thr1}$ and $Trust_{thr2}$), as follows:

- If (TT or $T_{n,t} < Trust_{thr1}$): untrustworthy elected neighbors, which necessitate should be blocked or eliminated from the network's operations.
- If (TT or $T_{n,t} > Trust_{thr2}$): trustworthy elected neighbors, which can be considered for selection as CHs.
- If ($Trust_{thr1} <$ TT or $T_{n,t} < Trust_{thr2}$): elected neighbors of the node with the acceptable trust values; nonetheless, these neighboring nodes may only play the role of the cluster's members (not CH).

H. *CHs' election and clusters' formation*

In this step, the node selects its most trustworthy elected neighbor as the CH and subsequently notifies this decision to other members of the cluster. For this purpose, the node systematically ranks its trustworthy elected neighbors based on their TT or $T_{n,t}$ values in descending order; consequently, the neighbor with the superior TT or $T_{n,t}$ value is designated as the CH (CHs' election stage). Afterthat, the CH determines the members of its cluster; given that the CH knows the trust values associated with the neighboring nodes aspiring to join this cluster, it systematically evaluates the trustworthiness of each such node. If a node's trust value falls below a predetermined threshold value, the CH will not permit that node to join its cluster; else, it will permit the node to become a member of its cluster if the trust value meets or exceeds the predetermined threshold value (clusters' formation stage). Therefore, at the end of this step, each node identifies and verifies its corresponding CH (for the purpose of requesting the requisite services), while each CH also identifies and validates the members of its cluster (for the provision of the requested services); this results in the establishment of trust-based clustering in the WSN (initially involving CHs' election followed by clusters' formation).

I. *Requesting the required services and replying to the requested services (optional)*

At this juncture, the members of the cluster request the necessary services, and the corresponding CH fulfills the requests by providing the requisite services.

J. *Maintenance and repetition*

Following a specified interval of the network activity, alterations in the CHs and clusters will occur and the CHs and clusters will be changed. In this step, each node reiterates the aforementioned procedures and updates (either increasing or decreasing) the trust values of its neighboring nodes at each interval. This is necessitated by the dynamic nature of trust, as the TSC-WSN is attentive to the behavioral patterns of nodes across varying time intervals.

IV. Results, Analysis, and Discussions: A Comparative Analysis of the Performance of TSC-WSN, Zhao et al., and Zeng et al. Clustering Algorithms

This section elucidates the results derived from the simulations conducted via TRMSim-WSN and NS2 simulators, alongside the theoretical statistical assessments utilizing Expert Choice and Grey Relationship Analysis tools. Following tables (Table 3-Table 5) and figures (Figure 4-Figure 17), provide a comparative analysis of the performance of the TSC-WSN against the clustering algorithms proposed by Zhao et al. [33] and Zeng et al. [34] The findings from both the simulations and statistical analysis reveal that the TSC-WSN is improved in terms of accuracy regarding the identification of untrusted nodes, scalability, fault tolerance, and the average of packet loss rate. Nevertheless, its performance is adversely degraded concerning resource consumption (including energy, bandwidth, and memory), traffic and processing overhead, computational complexities, and execution speed.

Table 3. Characteristics of the simulation environment

No.	Characteristic	Value
1	Number of executions	50
2	Number of networks	20
3	Number of sensor nodes (min, max)	(100, 200)
4	Average number of neighbor nodes	10
5	Percentage of clients	70 %
6	Percentage of the relay server	10 %
7	Percentage of malicious servers	20 %
8	Radio range	10

Table 4. Results of simulations: normalized functional values in terms of the positive evaluation criteria

No.	Positive criteria	Weights of criteria	TSC-WSN	Zhao et al.	Zeng et al.
1	Accuracy of detecting untrusted nodes	0.316	0.473	0.323	0.204
2	Scalability	0.281	0.360	0.338	0.302
3	Fault tolerance	0.228	0.360	0.340	0.300
4	Execution speed	0.175	0.273	0.333	0.393
	Overall performance	1	0.375	0.334	0.292

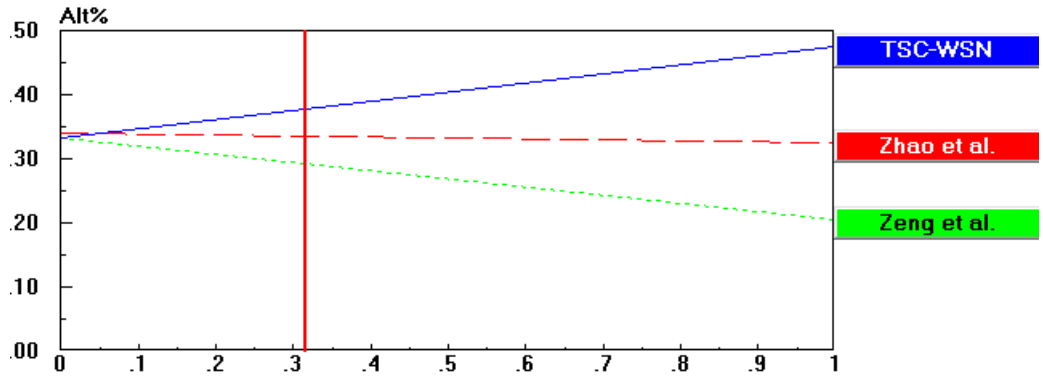


Figure 4. A comparative analysis of the performance of TSC-WSN, and the clustering algorithms proposed by Zhao et al. and Zeng et al., in relation to the accuracy of detecting untrusted nodes

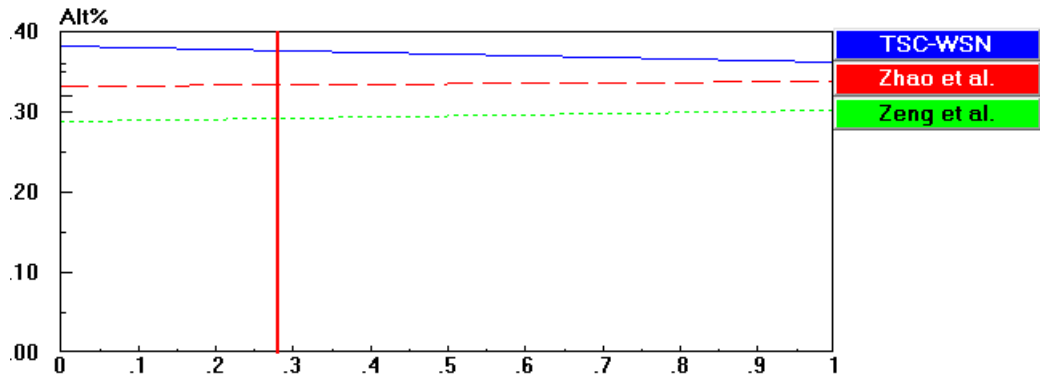


Figure 5. A comparative analysis of the performance of TSC-WSN, and the clustering algorithms proposed by Zhao et al. and Zeng et al., in relation to the scalability

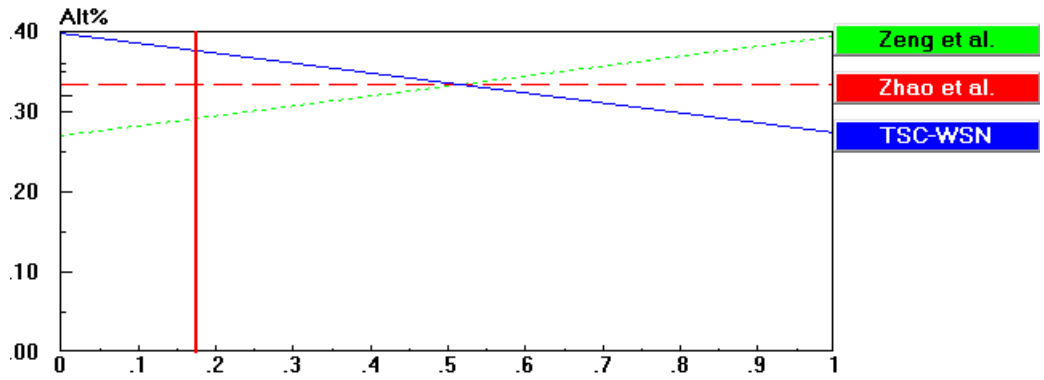


Figure 6. A comparative analysis of the performance of TSC-WSN, and the clustering algorithms proposed by Zhao et al. and Zeng et al., in relation to the execution speed

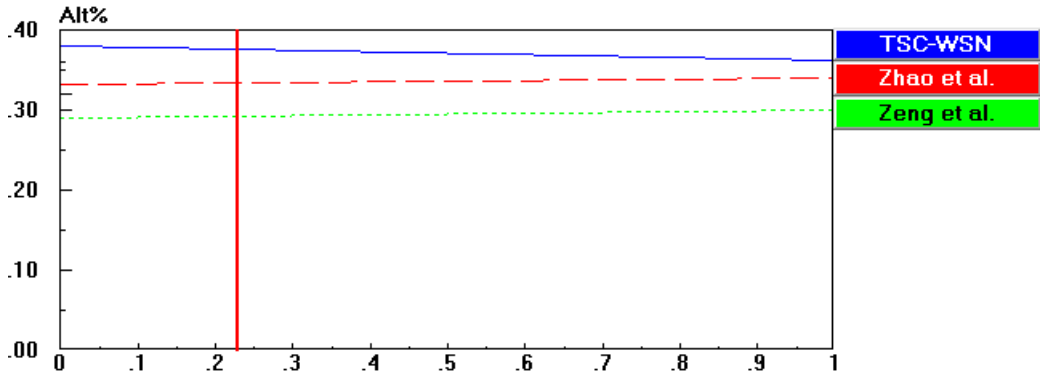


Figure 7. A comparative analysis of the performance of TSC-WSN, and the clustering algorithms proposed by Zhao et al. and Zeng et al., in relation to the fault tolerance

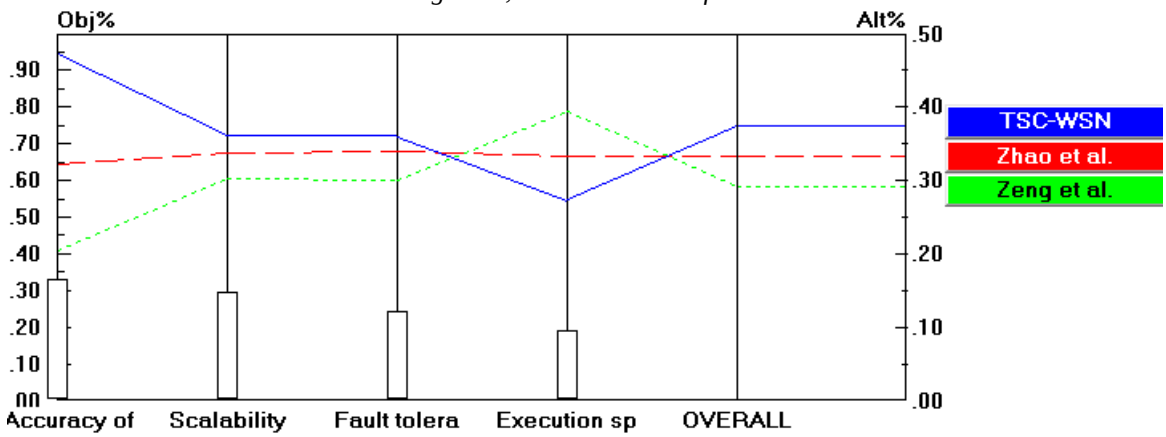


Figure 8. A comparative analysis of the performance of TSC-WSN, alongside the clustering algorithms developed by Zhao et al. and Zeng et al., based on the positive evaluation criteria

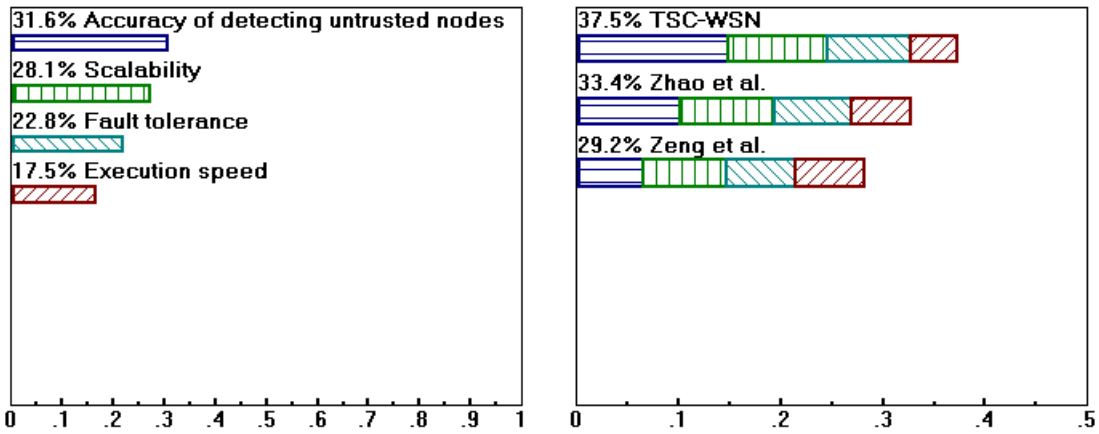


Figure 9. Sensitivity analysis regarding the dynamic behavior of TSC-WSN, Zhao et al., and Zeng et al. clustering algorithms as measured by the positive evaluation criteria

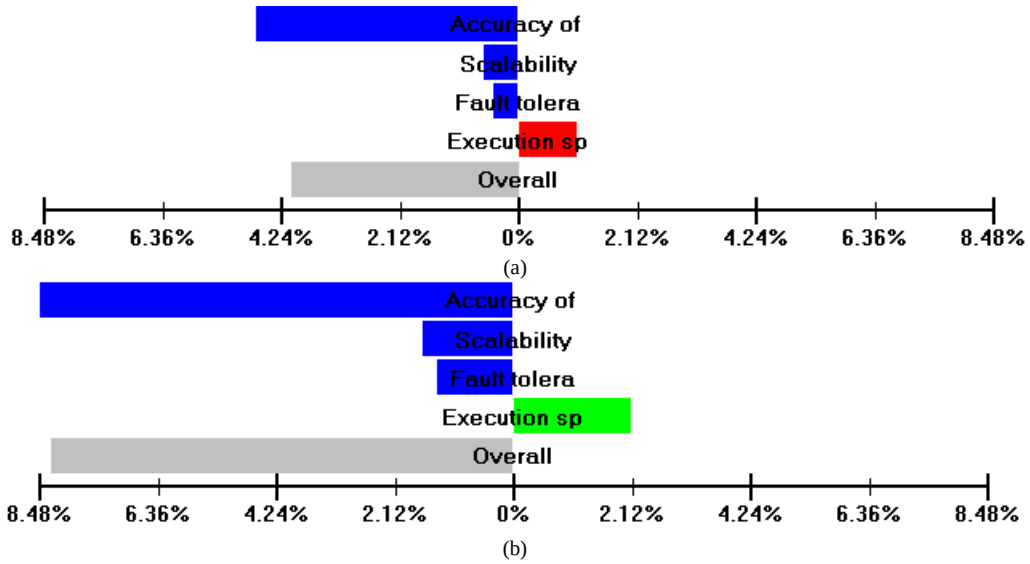


Figure 10. Weighted comparative sensitivity analysis: (a) between the TSC-WSN and Zhao et al. clustering algorithm and (b) between the TSC-WSN and Zeng et al. clustering algorithm with respect to the positive evaluation criteria

Table 5. Results of simulations: normalized functional metrics in accordance with the negative evaluation criteria

No.	Negative criteria	Weights of criteria	TSC-WSN	Zhao et al.	Zeng et al.
1	Resource consumption (energy, bandwidth, and memory)	0.299	0.383	0.328	0.289
2	Traffic overhead	0.266	0.386	0.335	0.278
3	Average of packet loss rate	0.243	0.238	0.357	0.405
4	Computational complexity and processing overhead	0.193	0.386	0.333	0.281
	Overall performance	1	0.351	0.338	0.312

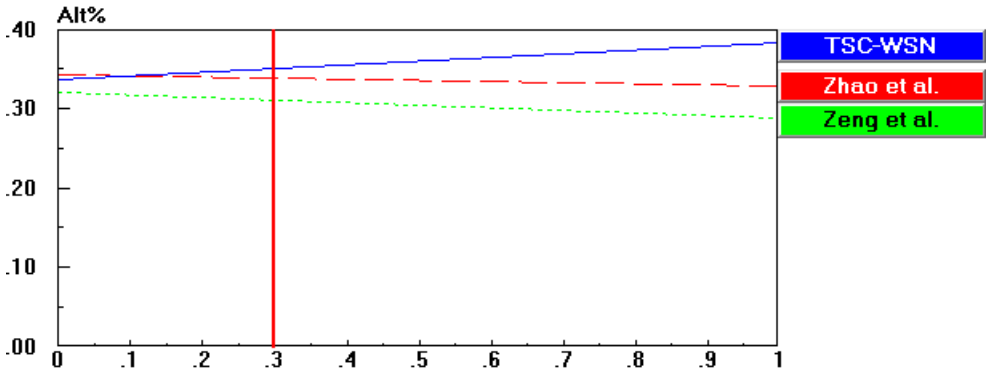


Figure 11. A comparative analysis of the performance of TSC-WSN, Zhao et al., and Zeng et al. clustering algorithms in terms of the resource consumption

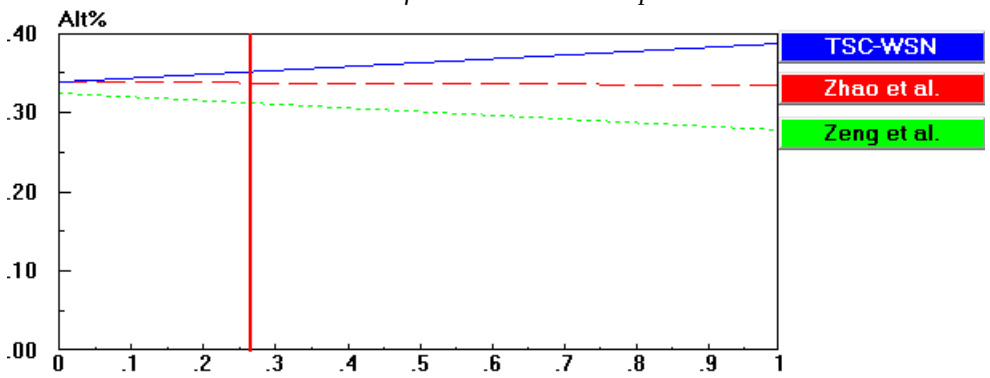


Figure 12. A comparative analysis of the performance of TSC-WSN, Zhao et al., and Zeng et al. clustering algorithms in terms of the traffic overhead

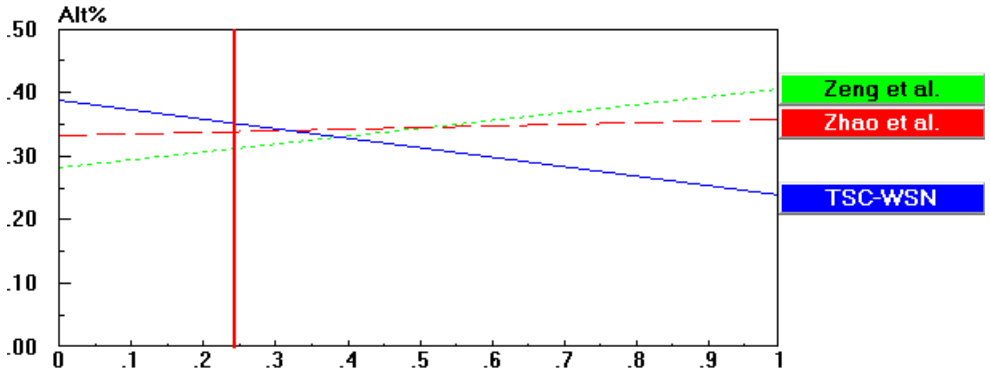


Figure 13. A comparative analysis of the performance of TSC-WSN, Zhao et al., and Zeng et al. clustering algorithms in terms of the average of packet loss rate

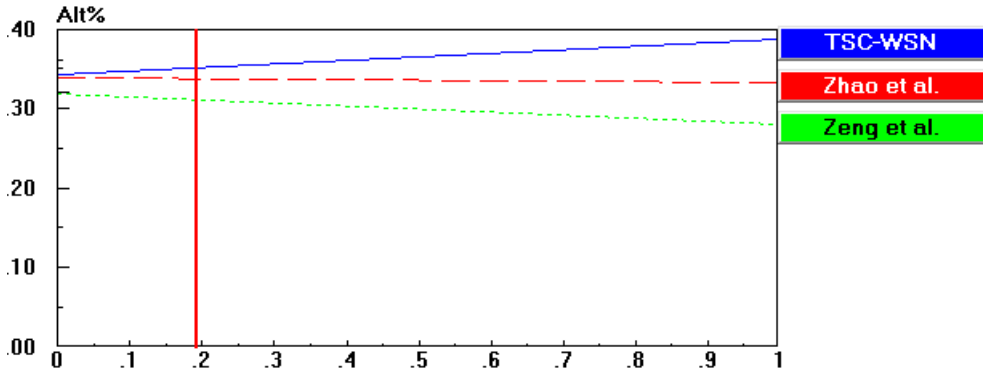


Figure 14. A comparative analysis of the performance of TSC-WSN, Zhao et al., and Zeng et al. clustering algorithms in terms of the computation complexity and processing overhead

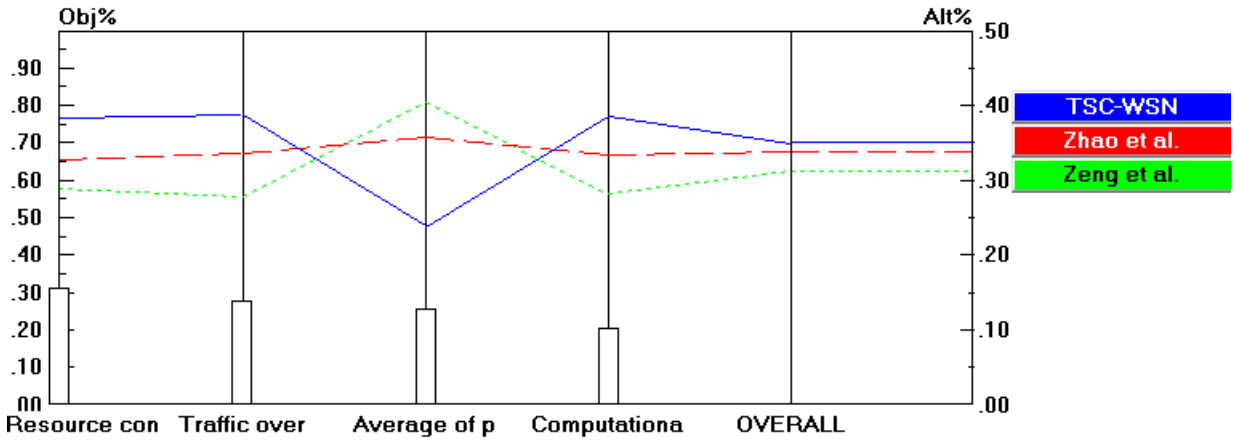


Figure 15. A comparative analysis of the performance of TSC-WSN, Zhao et al., and Zeng et al. clustering algorithms with respect to the negative evaluation criteria

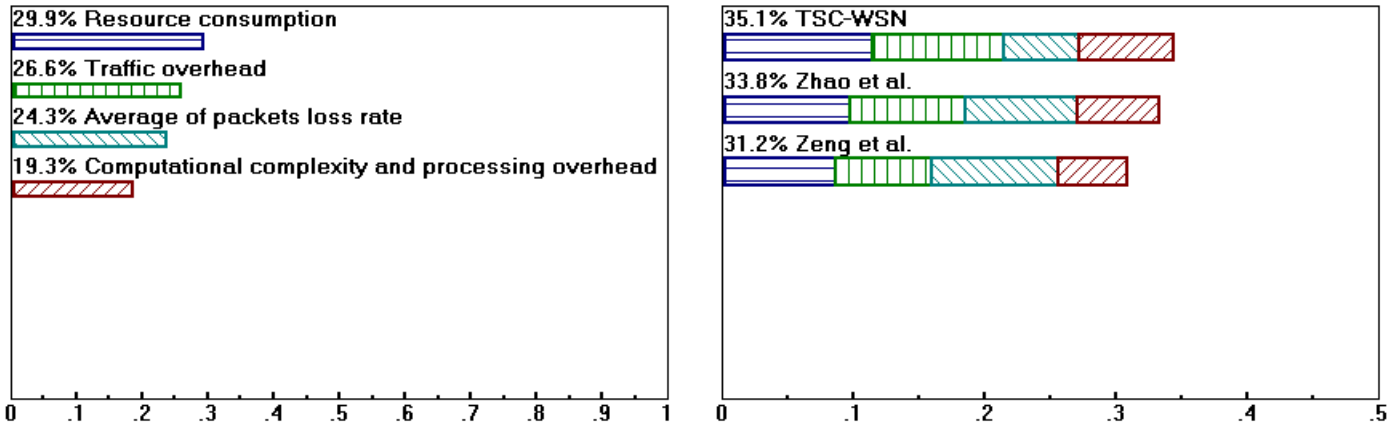


Figure 16. Sensitivity analysis regarding the dynamic behavior of TSC-WSN, Zhao et al., and Zeng et al. clustering algorithms as measured by the negative evaluation criteria

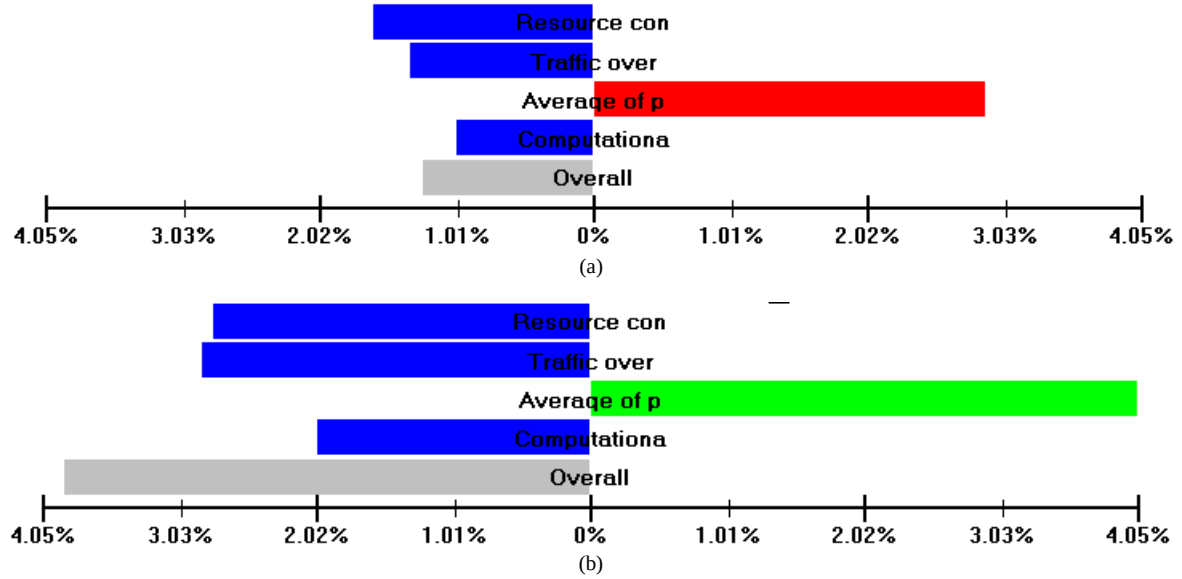


Figure 17. Weighted comparative sensitivity analysis: (a) between the TSC-WSN and Zhao et al. clustering algorithm and (b) between the TSC-WSN and Zeng et al. clustering algorithm in terms of the negative evaluation criteria

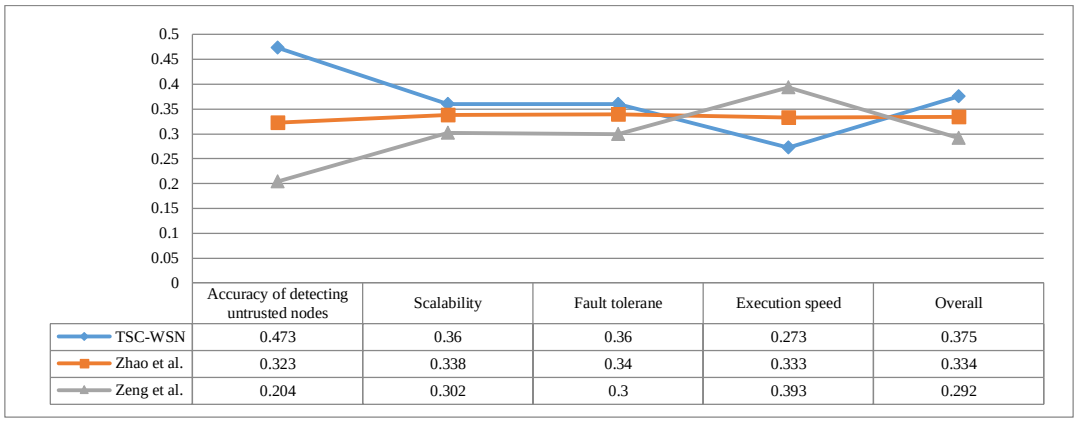
V. Conclusion and Future Directions

In the clustered WSNs, the process of trust-based clustering, encompassing trust-based CHs' election and trust-based clusters' formation, emerges as an important, essential, and complex research issue, due to it significantly affects the overall security and performance of these networks. Cryptographic security mechanisms are deemed inappropriate for WSNs; as they fail to detect a wide array of internal attacks and instances of selfishness behaviors; additionally, their implementation is characterized by significant costs and inconsistency with the distinctive features of WSNs. However, trust-aware security facilitated through the utilization of the TMSs and the establishment of trust relationships among nodes represents a significant security solution and constitutes a high-interest topic for further research. However, the current trust-aware clustering schemes have several deficiencies, including the imposition of substantial overhead on WSNs, computational and algorithmic complexities, excessive consumption of resources, low accuracy, inadequate security measures, high error rate in the identification of untrustworthy nodes, and no adaptability to the unique characteristics of WSNs, such as stringent resource constraints, high node density, dynamic network topology, and nodes mobility. Consequently, this research paper proposes and describes a trust-based clustering algorithm specifically designed for homogeneous WSNs, called TSC-WSN, which is adaptive to the distinctive characteristics of WSNs. It considers the trustworthiness of the nodes as the main criterion within the clustering process to enhance the network security. TSC-WSN is distinct from other clustering schemes concerning the criteria and procedures utilized for calculating and predicting the trustworthiness of nodes, electing the cluster heads, and forming the clusters. In continuation, Table 6 delineates some of the principal attributes of TSC-WSN.

Table 6. The principal attributes of the TSC-WSN: advantages and disadvantages

Properties	Description
Strengths	• Unpredictability, non-manipulability, and consensus-based election of CHs. • Distributed and highly scalable, with considerations for resource-awareness, congestion-awareness, dynamic and mobile nature, periodic-proactive strategies, multi-criteria approaches, and high flexibility. • The implementation of distributed computations and balanced resource consumption is facilitated by the dynamic rotation of the CH role among the members of the cluster. Additionally, each node is responsible for calculating and maintaining the trustworthiness of its designated neighbors within its local trust table. • High fault tolerance (there are no single points of failure). • The utilization of direct trust exclusively for the calculation and prediction of trustworthiness contributes to a reduction in the traffic and processing overhead, latency, computational complexities, and timely identification of untrustworthy nodes. • Security improvement through the prevention of false or forged data injections and packet dropping attacks via the identification the malicious, compromised, or selfish nodes by using the TMS, the selection of trusted stable nodes as CHs, the formation of trust-based clusters, and the dynamic rotation of the CH role across varying time intervals. • Security improvement through the resistance to attacks such as badmouthing and collusion of nodes (characterized by unfair evaluation and false commentary) is facilitated by using only direct trust calculations during the trustworthiness evaluation process; additionally, the impact of history clearing, water washing, and newcomer attacks is mitigated by assigning the low trust values to newly arrived nodes and applying diminished weight to previous trust values. • Deterministic-nature (non probabilistic) and topology-independent.
Weaknesses	• Security vulnerabilities like Sybil attack. • A plethora of criteria and monitored behaviors for trustworthiness evaluation lead to high traffic and processing overhead, resource wastage, and delays within the network's operations. • High memory consumption is necessitated by the maintaining maintenance of various decision matrixes, trust tables, and requisite computational memory. • The exclusive reliance on direct trust for the calculation and forecasting of trustworthiness results in a reduction in overall precision.

In accordance with Figure 18, Figure 19, Table 7, and Table 8, the performance of TSC-WSN is juxtaposed against that of the clustering algorithms developed by Zhao et al. and Zeng et al.; the results derived from simulations and statistical analysis indicate that TSC-WSN demonstrates improvements in parameters such as accuracy in detecting untrusted nodes, scalability, fault tolerance, and the average of packet loss rate. Conversely, its performance exhibits degradation in relation to parameters such as resource consumption, traffic and processing overhead, computational complexities, and execution speed.

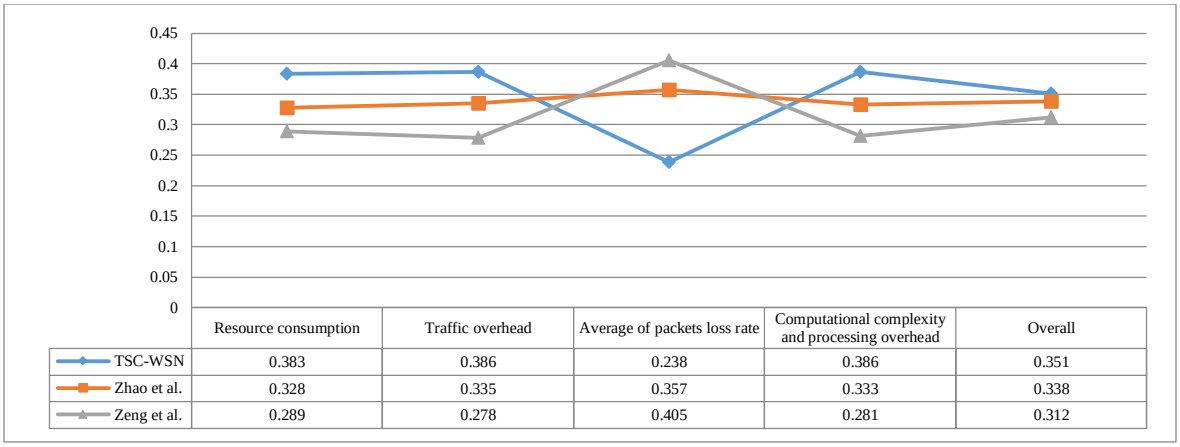


	Accuracy of detecting untrusted nodes	Scalability	Fault tolerane	Execution speed	Overall
TSC-WSN	0.473	0.36	0.36	0.273	0.375
Zhao et al.	0.323	0.338	0.34	0.333	0.334
Zeng et al.	0.204	0.302	0.3	0.393	0.292

Figure 18. A comparative evaluation of the performance of TSC-WSN, Zhao et al., and Zeng et al. clustering algorithms concerning the accuracy in detection of untrusted nodes, scalability, fault tolerance, and execution speed

Table 7. Percentage of Improvement in the performance of TSC-WSN relative to the performance of Zhao et al. and Zeng et al. clustering algorithms based on the positive evaluation criteria

No.	Positive criteria	Zhao et al.	Zeng et al.
1	Accuracy of detecting untrusted nodes	+31.71 %	+56.87 %
2	Scalability	+6.11 %	+16.11 %
3	Fault tolerance	+5.56 %	+16.67 %
4	Execution speed	-18.02 %	-30.53 %
	Overall performance	+10.93 %	+22.13 %



	Resource consumption	Traffic overhead	Average of packets loss rate	Computational complexity and processing overhead	Overall
TSC-WSN	0.383	0.386	0.238	0.386	0.351
Zhao et al.	0.328	0.335	0.357	0.333	0.338
Zeng et al.	0.289	0.278	0.405	0.281	0.312

Figure 19. A comparative evaluation of the performance of TSC-WSN, Zhao et al., and Zeng et al. clustering algorithms concerning the resource consumption, traffic overhead, average of packet loss rate, and computational complexity and processing overhead

Table 8. Percentage of Improvement the performance of TSC-WSN relative to the performance of Zhao et al. and Zeng et al. clustering algorithms based on the negative evaluation criteria

No.	Negative criteria	Zhao et al.	Zeng et al.
1	Resource consumption (energy, bandwidth, and memory)	-14.36 %	-24.54 %
2	Traffic overhead	-13.21 %	-27.98 %
3	Average of packet loss rate	+33.33 %	+41.23 %
4	Computational complexity and processing overhead	-13.73 %	-27.20 %
	Overall performance	-3.70 %	-11.11 %

Several other issues that should be further studied in future research are:

- Security analysis of the TSC-WSN, aimed at identifying its security vulnerabilities, and implementing enhancements accordingly.
- An analysis of the impact of trust-updating interval on the operational performance of TSC-WSN, as well as its implications for the longevity of the WSN.
- A discourse on the optimal configuration of clusters and the equitable distribution of CHs to achieve optimal energy efficiency.
- The development of a trust-aware intrusion detection system (IDS) designed for wireless sensor networks (WSNs).
- Feasibility study of integrating the TSC-WSN into the hierarchical routing protocols within WSNs.
- TMSs should be designed across the different layers of the WSNs' protocol stack, as security has to be ensured in all WSNs' layers; this necessitates a thorough discussion on the trust-aware data aggregation, trust-aware routing, and trust-aware encoding and decoding within the different layers of the WSNs' protocol stack.

CERTIFICATE OF AUTHENTICITY AND AI-FREE CREATION

The author guarantees the authenticity and originality of this paper; also, the content of this paper has been created entirely by the author without the use of artificial intelligence tools or any automated systems.

REFERENCES

[1] J. Yick, B. Mukherjee and D. Ghosal; Wireless Sensor Network Survey; Elsevier's Computer Networks Journal, Vol. 52, pp. 2292-2330; 2008.
[2] H. Jadidoleslamy; A Comprehensive Comparison of Attacks in Wireless Sensor Networks; International Journal of Computer Communications and Networks; Vol. 4, No. 1; 2014.
[3] H. Jadidoleslamy; A Distributed and Hierarchical Intrusion Detection Architecture for Wireless Sensor Networks; International Journal of Network Security and its Applications; Vol. 3, No. 5, pp. 131-154; 2011.
[4] S. Mohammadi, R. A. Ebrahimi, and H. Jadidoleslamy; A Comparison of Routing Attacks on Wireless Sensor Networks; International Journal of Information Assurance and Security; Vol. 6, pp. 195-215; 2011.
[5] H. Jadidoleslamy; A Novel Clustering Algorithm for Homogenous and Large-Scale Wireless Sensor Networks: Based on Sensor Nodes Deployment Location Coordinates; International Journal of Computer Science and Network Security; Vol. 14, No. 2; 2014.
[6] H. Jadidoleslamy, M. R. Aref, and H. Bahramgiri; A fuzzy fully distributed trust management system in wireless sensor networks; AEU International Journal of Electronics and Communications; Vol. 70, No. 1, pp. 40-49; 2016.
[7] H. Jadidoleslamy; TMS-HCW: a trust management system in hierarchical clustered wireless sensor networks; Security and Communication Networks; Vol. 8, No. 18; 2015.
[8] H. Jadidoleslamy, M. R. Aref, and H. Bahramgiri; A statistical distributed multipath routing protocol in wireless sensor networks; International Journal of Internet Protocol Technology; Vol. 9, No. 4, pp. 161- 173; 2016.
[9] H. Jadidoleslamy; A Hierarchical Multipath Routing Protocol in Clustered Wireless Sensor Networks; Wireless Personal Communications; Vol. 96, pp. 4217-4236; 2017.
[10] S. Soro and W. Heinzelman; Cluster head election techniques for coverage preservation in wireless sensor networks; Ad Hoc Networks; Vol. 7, No. 5, pp. 955–972; 2009.
[11] Y. Jin, L. Wang, Y. Kim, and X. Yang; EEMC: An energy-efficient multi-level clustering algorithm for large-scale wireless sensor networks; Computer Networks Journal; Vol. 52, pp. 542-562; 2008.
[12] C. Wen and W. Sethares; Automatic decentralized clustering for WSNs; EURASIP Journal on Wireless Communications and Networking; Vol. 5, No. 5, pp. 686-697; 2005.
[13] K. Yanagihara, J. Taketsugu, K. Fukui, S. Fukunaga, S. Hara, and K. I. Kitayama; EACLE: Energy-aware clustering scheme with transmission power control for sensor networks; Wireless Personal Communications; Vol. 40, No. 3, pp. 401-415; 2007.
[14] S. Yi, J. Heo, Y. Cho, and J. Hong; PEACH: Power-efficient and adaptive clustering hierarchy protocol for WSNs; Computer Networks; Vol. 30, pp. 2842-2852; 2007.
[15] J. Kamimura, N. Wakamiya and M. Murata; A distributed clustering method for energy-efficient data gathering in sensor networks; International Journal on Wireless and Mobile Computing; Vol. 1, No. 2, pp. 113-120; 2006.
[16] L. Qing, Q. Zhu, and M. Wang; Design of a distributed energy efficient clustering algorithm for heterogeneous wireless sensor networks; Computer Communications; Vol. 29, No. 12, pp. 2230-2237; 2006.
[17] A.M. Jubair, R. Hassan, A.H.M. Aman, H. Sallehudin, Z.G. Al-Mekhlafi, B.A. Mohammed, and M.S. Alsaffar; Optimization of Clustering in Wireless Sensor Networks: Techniques and Protocols; Applied Sciences; Vol. 11, No. 23; 2021.
[18] P. Rawat, and S. Chauhan S; A survey on clustering protocols in wireless sensor network: taxonomy, comparison, and future scope; Journal of Ambient Intelligence and Humanized Computing,14:3, pp. 1543-1589, 2023.
[19] A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen; Clustering objectives in wireless sensor networks: A survey and research direction analysis; Computer Networks; Vol. 180; 2020.
[20] F. Bao, I. R. Chen, M. J. Chang, and J. H. Cho; Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-aware Routing and Intrusion Detection; IEEE Transactions on Network and Service Management, Vol. 9, No. 2, pp. 169-183; 2012.
[21] T. Zahariadis, H. C. Leligou, P. Trakadas, and S. Voliotis; Trust management in wireless sensor networks; European Transactions on Telecommunications (Wiley); Vol. 21, Issue 4, pp. 386-395; 2010.
[22] G. Zhan, W. Shi, and J. Deng; TARF: A Trust-Aware Routing Framework for Wireless Sensor Networks; In Proceedings of 7th European Conference in Wireless Sensor Networks, Lecture Notes in Computer Science, Vol. 5970, pp. 65-80; 2010.
[23] C. Park, Y. Lee, H. Yoon, S. Jin, and D. Chio; Cluster based Trust Evaluation in Ad Hoc Networks; In Proceedings of 7th IEEE International Conference on Advanced Communication Technology; Vol. 1, pp. 503-507; 2005.
[24] G. V. Crosby, N. Pissinou and J. Gadze; A Framework for Trust-aware Cluster Head Election in Wireless Sensor Networks; In Proceedings of the Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems; pp. 10-22; 2006.
[25] B. Kadari, A. Mohamed and M. Feham; Secured Clustering Algorithm for Mobile Ad Hoc Networks; International Journal of Computer Science and Network Security; Vol. 7, No. 3, pp. 27-34; 2007.
[26] S. Peng, W. Jia, and G. Wang; Voting-aware Clustering Algorithm with Subjective Trust and Stability in Mobile Ad-Hoc Networks; IEEE/IFIP International Conference on Embedded and Ubiquitous Computing; pp. 3-9; 2008.
[27] F. Song and B. Zhao; Trust-aware LEACH Protocol for Wireless Sensor Networks; In Proceedings of the IEEE Second International Conference on Future Generation Communication and Networking; Vol. 1, pp. 202-207; 2008.

[28] R. Ferdous, V. Muthukkumarasamy, and E. Sithirasenan; Trust-aware Cluster head Selection Algorithm for Mobile Ad hoc Networks; International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11, pp. 589-596; 2011.
[29] T. Holczer and L. Buttyan; Anonymous aggregator election and data aggregation in wireless sensor networks; International Journal of Distributed Sensor Networks; Vol. 2011, pp. 1-19; 2011.
[30] I. Nishimura, T. Nagase, Y. Takehana and Y. Yoshioka; Secure Clustering for Building Certificate Management Nodes in Ad-Hoc Networks; In Proceeding of 14th International Conference on Network-aware Information Systems; pp. 685-689; 2011.
[31] G. Wang and G. Cho; Securing Cluster Formation and Cluster Head Elections in Wireless Sensor Networks; International Journal of Communication Networks and Information Security; Vol. 6, No. 1, pp. 70-88; 2014.
[32] M. Sheik Dawood, S. Sakena Benazer, S.K. Vijaya Saravanan, and V. Karthik; Energy efficient distance based clustering protocol for heterogeneous wireless sensor networks; Materials Today: Proceedings; Vol. 45, pp. 2599–2602; 2021.
[33] L. Zhao, S. Qu, and Y. Yi; A modified cluster-head selection algorithm in wireless sensor networks based on LEACH; EURASIP Journal on Wireless Communications and Networking; 2018.
[34] B. Zeng, Sh. Li, and X. Gao; Threshold-driven K-means sector clustering algorithm for wireless sensor networks; EURASIP Journal on Wireless Communications and Networking; Vol. 2024, Iss. 1; 2024.
[35] H. Hu, X. Fan, and Ch. Wang; Energy efficient clustering and routing protocol based on quantum particle swarm optimization and fuzzy logic for wireless sensor networks; Scientific Reports (Nature Publisher Group); Vol. 14, Iss. 1; 2024.
[36] R. Rajalingam, and K. Kavitha; Energy-Recognition Clustering Technique Based on Reinforcement Learning In WSN; Journal of Electrical Systems; Vol. 20, Iss. 2; 2024.
[37] A. I. Al-Sulaifanie, and B. K. Al-Sulaifanie; Hybrid access and adaptive duty cycle clustering protocol for ultra-low power wireless sensor networks; IET Communications; Vol. 15, Iss. 9; 2021.
[38] P. S. Mann, and S. Singh; Artificial bee colony metaheuristic for energy-efficient clustering and routing in wireless sensor networks; Soft Computing; Vol. 21; 2017.
[39] V. Chauhan, and S. Soni; Energy aware unequal clustering algorithm with multi-hop routing via low degree relay nodes for wireless sensor networks; Journal of Ambient Intelligence Human Computing; Vol.12; 2021.

How to cite: Hossein Jadidoleslamy, **A Trust-based Clustering Algorithm in Homogeneous Wireless Sensor Networks,** Journal of Distributed Computing and Systems (JDCS), Vol 8, Issue 1, Pages 8-25, 2025.