Trusted Energy-Aware, Hierarchical Routing (TEAHR) for Wireless Sensor Networks

Vikas K. *, Charu Wahi , Bharat Bhushan Sagar , Manisha Manjul

Posted Date: 21 February 2025

doi: 10.20944/preprints202502.1662.v1

Keywords: Wireless Sensor Network, Trust management, Energy, Anomaly, Routing

 doi:10.20944/preprints202502.1662.v1

Article

Trusted Energy-Aware, Hierarchical Routing (TEAHR) for Wireless Sensor Networks

Vikas[1,*], Charu Wahi [2], Bharat Bhushan Sagar [3] and Manisha Manjul [4]

[1,2] Computer Science and Engineering, Birla Institute of Technology, Mesra, Jharkhand, India
[3] Computer Science and Engineering, Harcourt Butler Technical University, Kanpur, Uttar Pradesh, India
[4] Computer Science and Engineering, Delhi Skill and Entrepreneurship University, G.B. Pant Engineering, Delhi, India
* Correspondence: vikask1003@gmail.com

Abstract: These days, wireless sensor networks (WSNs) are expanding fast and are used in many fields such as healthcare, battlefields, etc. Depending upon the sensor, they are transmitting a few bytes to megabytes, which is a considerable amount of data in a short duration, so security is a significant issue while transferring the data. Furthermore, it is essential to solve security concerns while transferring data by secure routing in wireless sensor networks, which critically depend on energy consumption. Trust is an essential parameter of reliable and safe communication between sensor nodes in dynamic WSNs. By developing trust relationships between sensor nodes, trust management is a successful method to overcome these issues. In WSNs, trust estimation methods are primarily employed in order to improve confidence, trustworthiness, system performance, lifecycle, decision-making processes, and relationships of cooperation among sensor nodes. The fundamental requirements of WSN are energy, connectivity, extended lifetime of the network, the availability of nodes, exchange of information, and memory overheads. In this paper, we present a novel trust-based algorithm to access the level of Trust among nodes called Trust-Based Energy-Aware Hierarchical Routing (TEAHR) in order to foster more reliable and energy-efficient WSNs. Several trust metrics like energy trust, forwarding Trust, consistency Trust behaviour and much more these ratings help in identifying potential malicious nodes (anomaly), hence ensuring network availability. With the exception of large-scale networks, we have performed a comparative study on another algorithm as well and demonstrated that the proposed model exhibits lower latency when compared to other models in terms of energy consumption (more significant) and the packet forwarding rate/lower. The comparisons of TEAHR with conventional techniques show that the proposed algorithm reduces total latency by 15%, enhances energy efficiency by around 20%, and maintains a stable packet forwarding rate, which is highly desirable for accurate operation in adversarial environments, as demonstrated through comparative analysis.

Keywords: Wireless Sensor Network, Trust management, Energy, Anomaly, Routing

1. Introduction

Trust, a fundamental component of human interactions, serves as a conceptual linchpin for establishing secure and reliable communication among sensor nodes. In WSN, Trust assesses node behaviour and encompasses a multifaceted evaluation of communication patterns, data integrity, and overall reliability. Integrating trust models provides an upward route to defend WSNs from malicious features, generates fidelity of transmitted messages and grows the life span of the network in operation. Abstract With the dramatic proliferation of wireless sensor networks, much is expected to provide a broad range of services and applications that come up with an overwhelming need for cost-effective, robust security solutions. Conventional WSNs, despite being irreplaceable in terms of their data-acquisition potential, are subject to limitations such as node compromise, which is tempered with data and communication interruptions. The vulnerabilities, escalated by the inherent

 doi:10.20944/preprints202502.1662.v1

resource constraints of sensor nodes, point towards a dire need for new ways to enhance the dependability and efficiency of wireless sensor networks. WSNs are naturally vulnerable because wireless communication is open, so we see many security threats. The concept of trust management for WSNs provides an easy and straightforward solution to couple the security and cooperation problem, which measures the degree/level by which sensor nodes in improving network performance are being trusted or distrusted. WSNs are capable of providing multi-tiered detection capability as they can gather information wirelessly and communicate. However, there are also security flaws in their open messaging, unattended deployment, and limited resources. Malicious nodes (compromised or faulty) can inject erroneous data, disrupt routing protocols, and consume energy, thereby jeopardizing the network integrity and application performance.

Recent studies have proposed various trust evaluation models for assessing the trustworthiness of sensor nodes. Z. Ye. et al.[1]have proposed a dynamic trust evaluation model (DTEM), considering several factors of communication, called communication trust, data storage, and processing demands for adaptively evaluating data flow quality, which is referred to here for DTEM as energy-aware reputation-based scheduling in CSPs. This model calculates direct Trust and evaluates indirect Trust using trusted third-party recommendations. The integration of direct and indirect Trust was achieved by assigning and combining dynamic weights. Trust-based security mechanisms involve the development of lightweight security protocols tailored to the resource-constrained environments of WSNs. These are trust-based cluster head election mechanisms, location-aware trust-based mechanisms for detecting and isolating compromised nodes, and scalable enforcement algorithms using many-to-one balanced evolutionary game theory. Mathematical models underpin these techniques and provide a theoretical foundation for trust management systems. Trust management systems are comprehensive frameworks that include the collection, storage, modelling, transferring, and decision-making of trust-related information[2]. They are designed to detect and defend against internal attacks such as DoS, bad-mouthing, and on-off attacks, among others. Trust models offer an appropriate mechanism for formally setting up and handling trust relationships among sensor nodes. The Trust of sensor nodes is evaluated in many dimensions, including data trust, behaviour trust, comprehensive Trust, and historical trust[3]. Relay nodes are also subjected to trust evaluation to ensure the integrity of the data transmission. A trust list was maintained to keep track of the trustworthiness of nodes within the network. Trust in wireless sensor networks is always considered a broader concept than just the evaluation of single-node behaviour. This refers to communication patterns, data correctness verification and collaborative decision-making. Through the amalgamation of trust metrics, anomaly detection algorithms, and cryptographic techniques, a comprehensive trust-based framework has emerged, which is promising for significantly enhancing the security posture of wireless sensor networks.

In WSNs, trust management requires customization of applications as trust management (TM) is challenging owing to the nature of WSNs, deployment of sensor nodes, and how the dynamic characteristics of sensor nodes change over time. The deployed sensor nodes create numerous complications in TM[4]. Hostile surroundings can harm or physically impact a sensor node, causing it to function poorly. In addition, sensors are vulnerable to physical capture and temptation because of the distant and unsupervised nature of the environment. This results in a node behaving in an intelligent but incorrect manner, further complicating this issue. The network's topology is dynamic, causing the neighbourhood to vary dynamically. It is crucial to manufacture sensors at a low cost, which results in limitations in their computational power, communication range, and battery capacity. Therefore, in light of these aspects, it is essential to develop trust estimations (TEs) that are effective in terms of energy and processing resources to align with the limitations of sensor nodes while also ensuring that they are strong enough to fulfil the security needs of TM. Furthermore, exchanging trust values between nodes may be restricted because of the increased energy consumption and congestion in the limited bandwidth caused by the message overhead. Proposing

 doi:10.20944/preprints202502.1662.v1

a trademark while considering the aforementioned considerations can be challenging. Executing TE and managing each Node in a large number of neighbouring nodes is a tedious task. Despite numerous proposed studies, current computational techniques need a comprehensive analysis of all essential design factors for TM. Furthermore, while TM is still in its early stages, numerous research difficulties require attention, particularly the confusion between TM, reputation management, and trust establishment.

To ensure secure communication in WSNs, it is essential to ensure that the intermediate nodes responsible for forwarding data packets are trustworthy within the network. Hence, an effective trust model needs to be created. The Trust Model is such that by interacting with its neighbours, each sensor node measures the quality of the Neighbouring Node. The trust model is one of the critical issues in WSNs. Various studies have also shown that a trust management system is a beneficial approach to identification and protection from threats[5]. Moreover, to avoid captured/compromised nodes from internal attacks, the trust model used here, the author has designed a trust management system. Trust management systems are employed to assess the attributes of information, provide network reliability services such as authentication and sensing the rogue nodes, and facilitate secure resource sharing. This is a more profound research area in understanding Trust Management technology mechanisms, designs and objectives of Trust management for WSNs, along with related work and open challenges. To find Trust, we need to cover the fundamental concepts of Trust, i.e. ideas and design elements. To construct a Trust Establishment (TE) scheme appropriately, it is crucial to have a thorough understanding of the fundamentals, i.e. definition, values, and properties of Trust. Furthermore, TM-related attacks, protection methods, and applications are discussed to provide a comprehensive understanding of TM. Understanding the concept, values, and features of Trust is crucial for estimating the trust levels of nodes. The trust model establishes and manages trust connections between two nodes to ensure that legitimate nodes may be trusted to participate in the information transmission process[2]. Trust is the level of belief in the future actions of other nodes as determined by prior experiences and observations of their behaviour. Trust in WSNs is as follows: The Trust that Node A has in Node B can be defined as the confidence, expectation, or assurance that Node B will be good in a manner that is sincere, competent, and honest in an upcoming activity or behaviour. Trust value: A wide variety of forms and ranges are used, which can be in the range of 0 or 1, 0 to 100; the form can be discrete or continuous. Trust properties: Trust properties are yet another essential component of both trust estimation and establishment. We consider the following trust properties: subjective attribute indicates that multiple trust values can be assigned to the same Node. The current degree of Trust may be influenced by past experiences. Continually changing leads to the possibility that trust values increase or decrease over time. The evaluation of Trust is influenced by the researcher's subjective viewpoint, and there is no consensus on the specific traits that should be included. In previous studies, the authors in[6] categorized trust as Trust relationships built on context attributes that encompass information about the relevant participant's situation. It is challenging to assess and monitor trust characteristics. These traits are related to cognitive and social Trust. Trust factors that may be assessed and tracked. These qualities focus on computing Trust. The authors [7] have defined trust qualities into numerous attributes, such as Trust is not 100%, i.e. one object cannot trust another object completely. The degree of Trust between nodes R and V and between nodes R and G is related to the transferability of Trust. This crucial quality can be used to reason complex relationships. In the past, researchers have proposed various trust management approaches for WSNs, each with its strengths and limitations. WSNs are increasingly becoming a critical component of modern technology infrastructure. WSNs have evolved into a fundamental technology for collecting and transmitting data in a variety of settings. However, the decentralized and often unattended nature of these networks makes them particularly vulnerable to security threats. There are many challenges, and trust management is essential for ensuring the reliability, security, and integrity of the data and the network itself. The deployment of WSNs in open, often hostile environments poses unique challenges that differ from traditional wired or centralized

 doi:10.20944/preprints202502.1662.v1

networks. The critical characteristics of WSNs include limited power resources, constrained computational capabilities, and the absence of a fixed infrastructure. These characteristics make WSNs highly susceptible to a range of attacks, such as data tampering, eavesdropping, and node capture. Furthermore, the ad hoc nature of these networks means that nodes must often rely on each other for routing data, making the network's overall security dependent on the trustworthiness of individual nodes.

The motivation for selecting trust management in WSNs is also driven by the need to choose an approach that aligns with the specific requirements of the network and its applications. In trust-based routing protocols, the trustworthiness of nodes is considered when selecting routes for data transmission, considering this approach helps ensure that data is transmitted through reliable nodes, thus reducing the risk of data being intercepted or tampered with. Another approach is reputation systems, in which nodes monitor the behaviour of their neighbours and assign reputation scores based on their observed behaviour. Nodes with low reputation scores are considered untrustworthy and may be excluded from participating in the network. When you need to identify malicious behaviour, you have to consider context-aware trust management systems, which are useful in dynamic environments where the behaviour of nodes may vary depending on external factors such as environmental conditions or the presence of interference. By considering the context, these systems can make more accurate trust assessments. So, for WSNs, trust management is an essential focus for researchers and practitioners alike. By selecting trust management, WSNs can achieve higher levels of security, resilience, and efficiency, enabling them to support a wide range of critical applications across different domains. Trust is a dynamic concept, and it is possible for it to exist for some time. It is inevitable that the level of confidence shifts over time. WSNs are assailable to several types of attacks. Therefore, it is necessary to create a robust and secure design and use secure algorithms. This paper is further divided into the following ways: in section 2, we discuss related work. In section 3, we focus on the proposed work and the discussion; section 4 discusses the results, and section 5 is the conclusion of the paper.

2. Related Work

As security is one of the major concerns apart from energy, data transfer, unreliable communication, etc. Figure. 1 displays the attack's kind in WSN using TM. We can classify the attacks in two ways, i.e. direct and indirect. In direct attack, the targeted attacks on the TM system result in a decline in TM system performance and cause inaccurate decision-making. Hence, the attack diminishes the efficiency of the network. In the context of secure routing, if the Traffic Manager (TM) is unable to identify hostile nodes owing to on-off or bad-mouthing assaults effectively, these malicious nodes can chosen as routing nodes for packet transmission. Therefore, the network experienced an elevated number of dropped packets. TM can be customized using abundant parameters, such as TE interval, rewarding and punishing processes, trust threshold, forgetting factor, dynamic trust parameter changes, and trust aggregation parameters. The optimal configuration of a Traffic Manager (TM) is contingent upon the specific use of TM, prevailing security conditions inside the network, and intended design objectives. Nevertheless, a TM system must meet specific fundamental criteria, including its ability to withstand rudimentary attacks. The purpose of trust management is to identify and protect against malicious node attacks to strengthen network security. For instance, if a malevolent node fails to transmit received data, the level of Trust diminishes.

 doi:10.20944/preprints202502.1662.v1

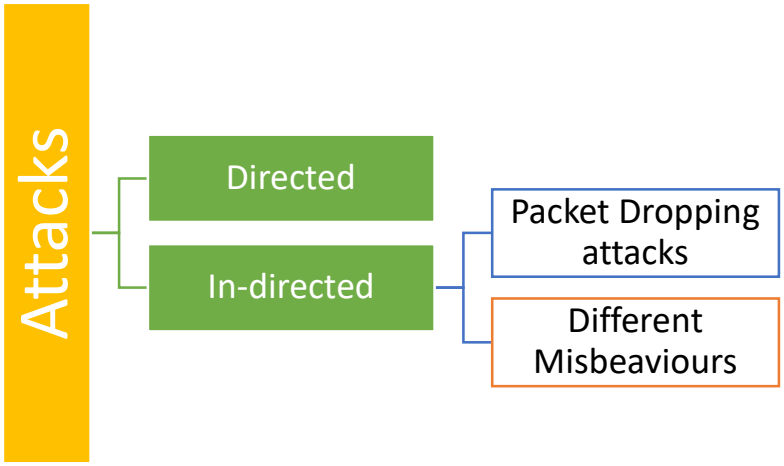

Figure 1. Attack in WSN.

In indirect attacks, the assaults do not specifically target the TM system, but the TM system can detect and prevent them. Such attacks may include deliberate data manipulation, depletion of energy resources, disruption of time synchronization, inaccurate reporting of sensor data, and several other malevolent behaviours. Addressing such threats and malicious behaviour are the primary goals of TM systems. A defence system against these attacks adheres to a standard process. Node performance was regularly monitored, observed, and documented. The defence procedure is mostly adjusted when employed to address various kinds of attacks. These attacks can be further classified as packet-dropping attacks in which malicious nodes attempt to impair the network operation by intentionally discarding packets using various methods, such as blackhole attacks, greyhole attacks and wormhole attacks. Another type is different misbehaviours. In general, nodes that misbehave can be categorized into two types: selfish nodes, which prioritize their own gain even if they harm others, and malevolent nodes, which intentionally undermine the performance of the system or other nodes without seeking personal benefits. Attacks can vary depending on the intentions of the misbehaving nodes. If a malicious node wants to deplete the energy of other nodes, it sends a substantial volume of traffic and forces the other nodes to respond. Another instance involves the transmission of forged, modified, or repeated routing information with the intention of causing routing loops, manipulating network traffic, altering source routes, generating false error messages, dividing the network, increasing the time it takes for data to travel from one end to another, and so forth. Owing to these attacks, the trust value was impacted.

2.1. Types of attacks

A malicious node can be identified by detecting its trust value in a timely manner. The comprehensive breakdown of the attacks is as follows: -

- Bad-mouthing attack [8]: This attack involves malicious nodes attempting to undermine the credibility of trustworthy nodes or enhance the credibility of malicious nodes through the dissemination of dishonest suggestions.
- Ballot stuffing attack[2]: A Confederate node enhances its reputation by supplying a substantial quantity of successful interaction data to the other side. To address such assaults, it is essential to decrease the weight of the indirect trust value offered by the neighbouring Node.
- On-off attack[9]: In an on-off assault, malicious nodes exhibit intermittent performance, alternating between periods of satisfactory and unsatisfactory behaviour. Malicious nodes can retain trust values even if they exhibit subpar performance. To effectively counter-switching attacks, older behavioural observations must carry a different level of significance than the more current behavioural observations.

 doi:10.20944/preprints202502.1662.v1

- Selfish attack[10]: The self-node will only delete the consent without reserving the resources to send the trust reply upon receipt of the trust request.
- Sybil attack[11]: The utilization of ID authentication and centralized trust models is a viable strategy to protect against Sybil attacks. These approaches not only enable the accurate identification of nodes but It also facilitates the detection of many false identities associated with malicious nodes through the network sink node or base station (BS).
- Sinkhole attack[12]: The assailant establishes a deceptive aggregation node to divert all data within the vicinity of the fraudulent sink node.
- Reputation time-varying attack[13]: A time-varying attack is a cyberattack strategy that changes its characteristics over time to evade detection and bypass security measures. These attacks exploit the vulnerabilities in systems designed to detect static attack patterns. By constantly changing their tactics, time-varying attacks can remain undetected for more extended periods, potentially causing more damage.

2.1. Trust Models and Their Classification

Recently, numerous schemes for trust management and reputation have been proposed, such as e-commerce, web-based services, peer-to-peer networks, and WSNs. It is essential to conduct a review of the most cutting-edge technology (TM) schemes that have been offered for WSNs to gain an understanding of the current status of the field and outstanding research concerns. In figure. 2, we present some proposed taxonomy schemes, comparisons, and TM schemes.

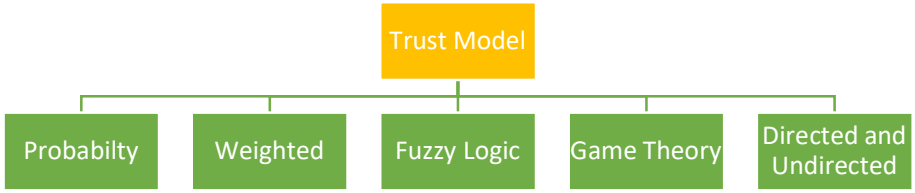

Figure 2. Trust Model[12.]

The probability trust model features are designed for security and reliability in WSNs. Trust models help in assessing the reliability of the sensor nodes and the data they transmit, which is crucial in environments where sensor readings are used to make critical decisions. They rely on probability theory to estimate the reliability of each Node in the network, including parameters such as its previous behaviour, recommendations from other nodes, and data provision quality. It is assumed that the sensor nodes will perform peer trust monitoring of each other, which represents a typical mode in practice for this problem under a probabilistic trust model. In a probabilistic trust model, which characterizes the usual approach to the issue in which each sensor node has its own monitoring trust score, it is expected that these sensors will behave or function correctly. This score is updated with time as the Node interacts with the network and the rest of the other nodes. The trust scores, if appropriately managed, can be used to decide whether a node is good enough for use in critical tasks or not, and if yes, then what type of queries should be answered by that particular Node while excluding others from the network based on their low trust levels. The resource-constrained nature of sensor nodes (e.g., power, memory and computational capacity) is one of the most challenging barriers to developing trust models in WSNs. Thus, trust models should be light and efficient when employed on WSNs. This kind of probabilistic trust model is beneficial in situations where we have to make decisions on their behaviour or the surroundings under uncertainties. It has a probabilistic strategy, and this makes the models capable of providing some estimation for Trust throughout the network, which is also vital while creating more secure WSNs. In WSNs, trust management typically relies on probabilistic models that use sensors' behaviour and interactions with other nodes in order to estimate the probability of the credibility for every Node. R. Feng et al. [14] proposed a trust model

 doi:10.20944/preprints202502.1662.v1

based on the Bayesian and entropy trust models. The Bayesian trust model updates the value based on the behaviour of observed nodes and utilizes Bayesian inference.

$$T_{new} = \frac{T_{old} \cdot P + S}{T_{old} + N} \quad (1)$$

T_{Anew} is the new trust value; T_{old} is the old trust value; P is the prior trust value; and S is the number of successful interactions. N is the total number of interactions. If we want to express the uncertainty in the Trust of some node, then this will be entropy. Therefore, lower entropy on the confidence of a node indicates that it is more reliable. The Trustfulness of a node, using Equation 2: The entropy (H) is defined based on the trust value in each Node;

$$H = -\sum_{i=1}^{n} p_i \, log \, x_2 \, p_i \quad (2)$$

p_i is the probability of the i-th outcome, $and\ n$ is the number of possible outcomes. WSNs warrant a trust management regime to ensure that the information collected and disseminated by nodes is accurate and reliable. A weighted trust model in WSN is used to evaluate and govern a node's credibility, which reserves its activities as well as interaction. The most appropriate model for this version is one that consists of components, each rated according to core and complementary networks. Trust metrics, trust evaluation, trust update, and trust-based decision-making are the significant components of the weighting trust model. Trust metrics are characteristics that indicate the reliability of a sensor node that is resourceful. These may be the packet forwarding ratio, data reliability, energy level, and cooperation degree per node, where each quality indicator is weighted with an essential factor for Trust to calculate. In the probability trust model, we also need to consider the direct and indirect Trust. The direct observations and recommendations are used to calculate direct and indirect Trust. Overall, Trust (T) can be a weighted sum of the direct T_d and indirect T_i:

$$T = w_d \cdot T_d + w_i \cdot T_i \quad (3)$$

T_{Wd} is direct weight, and w_i is indirect weight. As in WSN, where the trusts aggregate from different sources, and finally, the trust level of the recommenders is considered in the weighted average:

$$T_{agg} = \frac{\sum_{j=1}^{m} w_j \cdot T_j}{\sum_{j=1}^{m} w_j} \quad (4)$$

T_{agg} is the aggregated trust value, T_j is the trust value from the j-th recommender, w_j is the weight assigned to the j-th recommender's trust value, and m is the number of recommenders. A simple approach is to model the trust score T of a node using a weighted-sum model.

$$T = \sum_{i=1}^{n} w_i \ * \ m_i \quad (5)$$

where w_i is the weight of ith metrics, and m_i is the value of the ith metrics. In order to track the relevance of trust scores, Tools like Trust Update mechanisms are highly required. Trust scores can adapt by making updates in their values on new interactions or behaviours and developing methods to forgive nodes as well as penalize based on the change of actions. Trust scores are used in certain network decisions, such as routing and data aggregation or node cooperation (these will be discussed later). Nodes with a high trust score should be given priority for particular jobs, whereas those that are untrusted might end up being quarantined or scrutinized further. C Wahi et al. [15] Trust-Oriented Routing in Mobile Ad Hoc Networks (MANET) proposed a trust-based routing framework for establishing routes with neighbouring nodes, increasing AODV security levels by using historical node data to calculate the cooperation level, including both direct and indirect recommendations, and recompenses cross-recommendations. It is confirmed by a sequence of tests that such a version improved over TSAODV, called original AODV and TSR, across all scenarios in the work, but significantly when different parameters regarding the total number of nodes and

 doi:10.20944/preprints202502.1662.v1

distribution node velocity are changed. Research on it strengthens the weight trust model and uses Machine Learning Algorithms (MLAs) that update weights depending on the network state, so this adaptivity improves the accuracy of trust evaluation. A crucial illustrative example was found in a series of studies by Alshehri et al. These advances are discussed by[16], which showed better network performance and security. Author Chen et al. Present behaviour trust, data trust, and Historical Trust. The comprehensive Trust is calculated through a weighted approach, where each type of Trust contributes to the overall trustworthiness of a node. The formula for comprehensive Trust could be represented as $T_{comprehensive} = w_1 \cdot T_{behaviour} + w_2 \cdot T_{data} + w_3 \cdot T_{historical}$, where w_1, w_2, and w_3 are the weights assigned to each trust component, respectively. Reddy et al. [17] have introduced a model that evaluates both communication trust and data trust. Communication trust is derived from direct and indirect observations of a node's forwarding behaviour, while data trust is computed using the median of sensor data. The direct Trust could be calculated as

$$T_{direct} = \frac{successful\ forward}{total\ forwards} \tag{6}$$

Moreover, the Indirect Trust uses the weighted Dempster–Shaffer theory. The authors Prabha et. al.[18] employed fuzzy logic to evaluate Trust by considering metrics such as message success rate, elapsed time at a node, correctness, and fairness. Ye et al.[1] proposed a dynamic trust evaluation model (DTEM) that dynamically adjusted the weights of direct and indirect Trust. Direct Trust includes communication, data, and energy trust, whereas indirect Trust is derived from trusted recommendations. The integrated Trust can be calculated as

$$T_{integrated} = \alpha . T_{direct} + (1 - \alpha) . T_{indirect} \tag{7}$$

Where α is the weighted factor dynamically adjusted based on network conditions, Prabha et al. [18] in this author propose a trust model based on fuzzy logic to assess the trustworthiness of nodes in WSNs by considering several metrics: the message success rate, elapsed time at a node, correctness, and fairness. The trust value in this model is calculated by the fuzzification of these input metrics and the application of fuzzy inference rules. Finally, defuzzification of the result is used to find the final trust level, which is usually further broken down into low, medium, or high. Although not explicitly stated in the paper, a typical step in most fuzzy logic systems is fuzzification, in which all crisp input values are changed into degrees of membership of fuzzy sets. For example, the message success rate can be placed under sets such as Low,' 'Medium,' and 'High. The trust value can be calculated using the membership function μ, as follows:

$$T = \frac{\mu_{High}(x)}{High} + \frac{\mu_{Medium}(x)}{Medium} + \frac{\mu_{Low}(x)}{Low} \tag{8}$$

Where x is an input parameter, such as the number of successful data transmissions by M. Patra et al.[19]. A Fuzzy inference rule in which IF-THEN rules are applied based on fuzzified inputs to determine the trust level. An example rule might be IF (Message Success Rate is High) AND (Elapsed Time is Low) THEN (Trust is High). The defuzzification converts the fuzzy output of the inference process into a crisp value that represents the final trust level of the Node.

$$Y_{crisp} = \frac{\sum_{i=1}^{n} \mu_Y(y_i) \cdot y_i}{\sum_{i=1}^{n} \mu_Y(y_i)} \tag{9}$$

Where $\mu_Y(y_i)$ denotes the membership value of the output variable at y_i D. Han et al.[20]. In trust aggregation, the trust values from different sources can be aggregated using fuzzy operators such as the fuzzy weighted average

$$T_{agg} = \frac{\sum_{i=1}^{k} w_i \cdot T_i}{\sum_{i=1}^{k} w_i} \tag{10}$$

Where T_i is the trust value from the i-th source, and wi is the weight assigned to it by R. Patel et. al. [21]. The overall Trust value T can be conceptualized as a function of the input metrics

 doi:10.20944/preprints202502.1662.v1

processed through the fuzzy logic system: T = *f* (Message Success Rate, Elapsed Time, Correctness, Fairness). The Game theory-based trust models in WSNs apply the principles of game theory to analyze and optimize interactions among sensor nodes, focusing on Trust. These models treat interactions as strategic games in which nodes make decisions on their objectives, constraints, and expected actions by other nodes. The objective is to design mechanisms that promote cooperative behaviour while discouraging the behaviour of malicious or selfish nodes, thereby improving the overall trustworthiness and reliability of the network. In such a trust model, under game theory, every sensor node represents a player. The available strategies for players include forwarding packets, participating in network operations, and conserving energy. The results from these strategies, owing to the actions of other nodes, determine trust levels and, thus, payoffs for the players. In principle, Trust can be modelled as a game by using a trust model.

$$\Gamma = (N, S, u) \tag{11}$$

where N is the set of players (sensor nodes), $S = S_1 * S_2 * \ldots\ldots * S_n$ is the set of strategy profiles, where s_i is the strategy set available for player I; $u = u_1 * u_2 * \ldots * u_n$ is the vector of payoff functions, with u_i $(s_1 * s_2 * \ldots * s_n)$ representing the payoff to player i when strategy profile $(s_1 * s_2 * \ldots\ldots * s_n)$ is employed. The level of Trust of a node can be derived from its payoff, which reflects the behaviour of this Node to achieve network objectives, such as good data transmission. The common approach to modelling this uses utility functions that capture the benefits and costs related to different strategies. A simplified utility function by Abdalzaher et al. [22] for a sensor node may be written as:

$$u_i\,(s_i, s-i) = \alpha * B\,(s_i, s-i) - \beta * C(s_i) \tag{12}$$

Where $u_i\,(s_i, s-i)$ is the utility for Node I choosing strategy $s_i$is given strategies *s-i* of other nodes; $B\,(s_i, s-i)$ represents the benefits obtained from choosing strategy s_i, which may depend on the actions of other nodes. $C(s_i)$ for costs incurred if strategy si is chosen, α and β are weighting factors that maintain a balance between the benefits and costs. Those who define game models often look for Nash equilibria where none of the players feel compelled to change their strategy independently, being convinced of the strategies of all the others. In this context, regarding Trust, a Nash Equilibrium by Abdalzaher et al. [23] outlines the situation in which all nodes act in such a way as to maximize their utility while maintaining a satisfactory degree of trustworthiness and cooperativeness. Most related work in this area has focused on developing sophisticated mechanisms for dynamically adaptive changes in networks. At the same time, it has been incorporated with realistic factors, such as node mobility, variable trust levels, and possible adversaries. An evolutionary game is the study of the strategy evolution dynamics in a population. M.Adnan et al. [24] proposed a WSNs model of how trust-related strategies evolve among sensor nodes, and for that, the replicator equation is defined

$$\dot{x}_i = x_i\big(U_i(x) - \bar{U}(x)\big) \tag{13}$$

Where $\dot{x}_i$ is the proportion of the population using strategy I, $U_i(x)$ is the rate of change of x_i, $U_i(x)$ is the utility of strategy I, *and* $\bar{U}(x)$) is the average utility of the population. The models of Trust in Wireless Sensor Networks (WSNs) are the directed graph-based trust model and the undirected graph-based trust model. In both cases, graph theory is utilized to show relationships and interactions among the sensor nodes, where levels of Trust are shown in the graph's edges. The nodes and relationships represent the vertices defined by the edges within these models. Directed Graphs, or simply Digraphs, are used when the trust relationship is said to be asymmetric. That is, the trustworthiness of A to B is not necessarily the same as that of B to A. On the other hand, in undirected graphs, symmetric trust relationships were presented. That is, Trust is mutual and equal in both directions. In directed graph-based models, each edge is associated with a given direction. Directed edges show the relation of Trust from one Node to another, usually labelled with a weight. Then, the edge may further be used to represent the level of confidence. Such a model could use the sum of all incoming edge weights as a measure of the trustworthiness of the Node:

 doi:10.20944/preprints202502.1662.v1

$$T(v) = \sum(u, v) \in Ew(u, v) \quad (14)$$

Here, T(v) is the trust value for mode v, E is the set of edges, and w(u,v) refers to the weight of the edge from node u to node v. In undirected graph-based models, the edges are not directed and represent the mutual trust relationship between the nodes. The level of Trust, in this case, can be calculated based on the average weight of the edges that it connects as:

$$T(v) = \deg(v)1\sum(v, u) \in Ew(v, u) \quad (15)$$

Where deg(v) is the degree of node v, and w(v,u) is the weight of the edge connected nodes v and u. This graph can also support extra features, such as history data, node behaviour, and social trust metrics. For instance, a dynamic weighted graph model may change the edge weights with time, according to node actions, following mathematical expressions, to gradually increase or decrease the level of Trust from positive or negative interaction. Also, some of the most recent studies have been using sophisticated techniques, such as machine learning, for predictions and updating trust levels in conjunction with graph-based models. For example, the study conducted by Wang et al.[25] presented a dynamic model of trust management based on the directed graph, where temporal and spatial parameters are taken into account to change the trust levels depending on network conditions and behaviours. In his work, Xinying Yu et al.[26] have established the Energy Trust Model (ETM), which includes the remaining energy of nodes and trust level to obtain secure communication paths that will consequently guarantee the confidentiality of transmitting data within a WSN. Table 1 summarises the above various types of trust management in WSN.

Table 1. Comparative Analysis of Trust Models.

Feature/ Aspect	Probability Trust Model	Weighted Trust Model	Fuzzy Logic Trust Model	Game Theory-Based Trust Model	Directed Graph-Based Trust Models	Undirected Graph-Based Trust Models
Trust Calculation Method	Updates trust based on Bayesian inference; uses past behaviour to adjust trust scores.	Aggregates various trust metrics with weighted importance factors	Uses fuzzy logic to combine various metrics; Trust is determined by fuzzy inference rules	Models trust as a game where nodes interact and adjust strategies based on outcomes	Trust is calculated by summing the weights of incoming edges from other nodes	Trust is calculated based on the average weight of connected edges
Inputs Considered	Previous trust values, successful interactions, and total interactions	Metrics like packet forwarding ratio, data reliability, energy levels	Message success rate, elapsed time, correctness, fairness	The payoff, node actions, utility functions	Weights of edges in the directed graph	Weights of edges in the undirected graph
Handling of Uncertainty	Incorporates prior probabilities to handle uncertainty	Handles uncertainty by adjusting weights dynamically	Fuzzy logic manages uncertainty by handling imprecise input values	Game theory models interactions under uncertainty and evolves strategies over time	Directly handles asymmetric trust relationships	Handles symmetric trust relationships
Applicability	Suitable for scenarios with historical behavior data	Best for networks where multiple trust metrics are relevant	Ideal for networks with complex, multifaceted trust metrics	Effective in dynamic and adversarial environments where	Useful in networks where Trust is not reciprocal or equal	Applicable in scenarios where mutual Trust is required

 doi:10.20944/preprints202502.1662.v1

				strategies evolve	between nodes	between nodes
Strengths	Adjusts well to new data, incorporates past behaviours	Flexible and adaptable with various trust metrics	Handles complex and vague data inputs effectively	Encourages cooperative behaviour, handles adversarial nodes well	Effective for modelling directed trust relationships	Provides a clear model for mutual Trust in networks
Weaknesses	It can be computationally intensive, requires a lot of historical data	Requires careful selection of weights, might be sensitive to metric importance	Complexity in defining fuzzy rules and membership functions	Requires sophisticated modelling, might be complex to implement	May not handle symmetric relationships well	It may not represent asymmetric trust relationships effectively

2.1. Hierarchical Routing

A large number of routing protocols have been proposed in WSN because of their inherent characteristics, as well as their application and architectural requirements. These routing protocols can be categorized based on the network structure, protocol operation, and path base. The selection of an appropriate hierarchical routing algorithm should consider various factors like network size, topology traffic patterns, and energy constraints. These are essential factors that need to be explicitly considered while choosing an algorithm for any application in WSN.

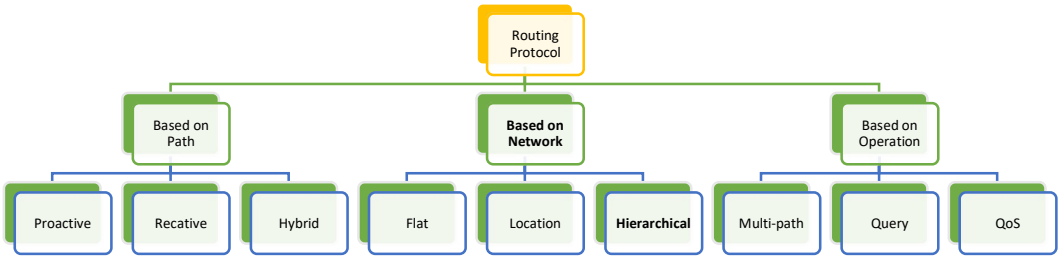

Figure 3. Routing Protocols in WSN[27.]

The hierarchical routing algorithm in WSNs is essential for reducing energy consumption and improving the transmission efficiency of large-scale deployments as well. Hierarchical algorithms arrange the nodes in a hierarchy and then partition the network into clusters or partitions. In the cluster, most packets are forwarded through direct communication with CH or trusted intermediate nodes if it is not possible to communicate directly. Inter-cluster Communication means when communication between clusters. When communicating outside the cluster, CH relays packets to other local(CH) or a base station. This will allow routing decisions based on Trust firmly to find the paths for both reliability and energy efficiency. Nodes whose trust levels fall below the minimum Threshold due to malicious actions or inconsistent behaviour can be isolated from critical network operations or removed from the network to maintain overall security and efficiency. By systematically managing Trust and continuously monitoring the behaviour of both existing and new cluster members, a WSN can effectively safeguard against malicious nodes (anomaly) and ensure reliable and secure network operation. Here are a few hierarchical routing algorithms that can be found in virtually all WSNs.

LEACH[28] (Low-Energy Adaptive Clustering Hierarchy) is a system in which nodes select the CH based on the probability threshold. Therefore, CHs aggregate data from their member nodes and

 doi:10.20944/preprints202502.1662.v1

transmit this to a sink node. LEACH is scalable and very easy to deploy, but it may need more energy consumption across nodes, especially near the sink. Power Efficient Gathering in Sensor Information Systems (PEGASIS) [29] is a token-passing mechanism in which each sensor node receives from and transmits to the nearest sensor node, and CH, having a token, will transmit it to BS; by doing so, conserve the energy. The token is rotated to make sure all sensor nodes participate in routing. PEGASIS is sensitive to node failure, especially CH. In this protocol, nodes closer to the sink node may consume more energy due to a more significant increase in transmission. Ad-hoc On-Demand Distance Vector Routing (AODV) [30]is not required in this global occurrence of advertising at specific intervals. Overall bandwidth is less required by each sensor node in this routing protocol. Apart from these, researchers have proposed many more routing algorithms.

2.1. Problem with existing schemes

The Trust-based algorithms have many inherent problems that can impact their performance, reliability, and overall effectiveness. Some of the problems with existing trust-based algorithms in WSNs are: If malicious nodes become intermediate nodes in data paths, they can manipulate trust scores by selectively forwarding or dropping packets, which causes anomalies in a sensor network. Trust-based algorithms often require significant computational, storage, and energy resources, which can be burdensome for sensor nodes with limited capabilities. Maintaining and updating trust scores involves frequent message exchanges, which can increase network traffic and thus consume additional energy and bandwidth. As the number of nodes in a network increases, the scalability of trust-based algorithms becomes problematic. The complexity of managing, updating, and evaluating the trust relationships increases, resulting in reduced efficiency and increased delays. In environments where the Node's topology is continuously changing, it is challenging to maintain updated trust relationships.

Proposed work

In WSNs, identifying compromised or malicious nodes that can disrupt data flow or instil harm is quintessential. Trust mechanisms evaluate node behaviour. As a result, Trusted Energy-Aware Hierarchical Routing (TEAHR) protocols are capable of addressing the following issues that WSNs have to encounter: Limited energy: sensor nodes are equipped with tiny batteries, so energy management must be taken care of cautiously. Network security: WSNs are threatened by many attacks. Trust mechanisms can resist malicious behaviour. Scalability: TEAHR protocols can efficiently cope with large-scale WSNs.

2.1. Research Contribution

In this paper, we proposed an energy-efficient framework to enhance the wireless sensor network security and detect anomaly detection. The TEAHR framework is designed to improve the security and performance of WSNs by dynamically assessing the trustworthiness of sensor nodes based on multiple criteria. The core idea behind TEAHR is to integrate trust evaluation with energy-aware routing to create a robust and efficient network. In the proposed framework, by considering factors such as energy consumption, packet forwarding effectiveness, consistency in behaviour, and adherence to network protocols, the TEAHR framework provides a comprehensive approach to managing Trust and enhancing the reliability of WSNs. The critical components of TEAHR are as follows:

- Energy Trust: Energy trust refers to the evaluation of a node's remaining energy levels as a measure of its operational reliability and expected lifespan. In WSNs, energy efficiency plays a critical role due to the limited power resources of sensor nodes, which are often powered by tiny batteries. Nodes with higher energy reserves are more likely to continue functioning reliably, making them more trustworthy within the network.

 doi:10.20944/preprints202502.1662.v1

- Forwarding Trust: Forwarding Trust assesses a node's effectiveness in forwarding packets to their intended destination. In a WSN, the ability of a node to reliably forward data is crucial for maintaining data integrity and ensuring that information reaches the intended recipients. A node's forwarding Trust is determined by monitoring its packet forwarding history, including metrics such as the number of successfully forwarded packets, packet loss rates, and transmission delays. Nodes that demonstrate consistency and reliability in forwarding behaviour are assigned higher trust scores, which make them more favourable for inclusion in routing paths.
- Consistency Trust: The level of consistency in a node's behaviour over time, particularly in relation to packet forwarding, is evaluated by consistency trust. This metric helps detect deviations that may signal potential malignancy or hardware failure. Inconsistencies in a node's behaviour, such as sudden drops in forwarding success rates or erratic energy consumption, can indicate that the Node has been compromised or is experiencing technical issues. TEAHR can identify nodes that may pose a risk to the network and adjust routing decisions accordingly to maintain overall network stability.
- Behavioural Trust: Behavioral Trust examines a node's adherence to network protocols and its past activity patterns to detect anomalies or potential security breaches. This metric is critical for identifying nodes that may be engaging in malicious activities, such as data tampering, unauthorized access, or collusion with other compromised nodes. Behavioral Trust is assessed by analyzing a node's interaction with other nodes, its response to protocol commands, and any deviations from expected behaviour. Nodes that exhibit suspicious or non-compliant behaviour are assigned lower trust scores and may be excluded from critical network functions.
- Anomaly Detection: a vital feature of the TEAHR framework, enabling the identification and isolation of malicious nodes within the network. When a node's trust score falls below a certain threshold due to suspicious behaviour, it is flagged as potentially compromised. The TEAHR system then takes proactive measures to manage the anomaly, such as restricting the Node's access to the network or isolating it entirely. This approach ensures that the impact of malicious nodes is minimized and that the network remains secure and operational. Additionally, the system continuously monitors the trust scores of all nodes, allowing for dynamic adjustments based on real-time behaviour.
- Energy Efficiency: TEAHR improves energy efficiency across the network by using energy metrics in trust computation. Nodes that are more energy-efficient are rated positively in trust evaluations, which incentivizes the use of energy-saving practices among nodes. This focus on energy efficiency not only extends the operational life of the network but also enhances its overall performance by reducing the likelihood of node failures due to energy depletion.

Advantages of the Proposed TEAHR Framework. The TEAHR framework offers several advantages over traditional trust management and routing protocols in WSNs:

- Enhanced Security: By incorporating multiple trust metrics, including energy, forwarding, consistency, and behavioural Trust, TEAHR provides a comprehensive approach to identifying and mitigating security threats. This multifaceted trust evaluation process makes it more difficult for malicious nodes to evade detection, thereby enhancing the overall security of the network.
- Optimized Energy Consumption: The integration of energy trust into the routing process ensures that nodes with sufficient energy reserves are prioritized, reducing the likelihood of network disruptions due to node failures. This focus on energy efficiency helps to extend the lifespan of the network and reduces the need for frequent maintenance or node replacement.
- Improved Network Reliability: TEAHR's emphasis on forwarding trust and consistency trust ensures that data is transmitted through reliable and consistent nodes, which enhances the integrity of the data and

 doi:10.20944/preprints202502.1662.v1

reduces the likelihood of packet loss or delays. This improved reliability is essential in critical applications where accurate and timely data transmission is essential.

- Scalability: The hierarchical nature of the TEAHR framework allows it to efficiently manage trust and routing decisions in large-scale WSNs. The system can scale to accommodate networks with a large number of nodes without compromising on performance or security, making it suitable for deployment in diverse and expansive environments.
- Dynamic Trust Management: TEAHR's ability to dynamically assess and adjust trust scores based on real-time behaviour ensures that the network can quickly respond to emerging threats and changes in node behaviour. This adaptability is crucial in environments where network conditions and node behaviour can change rapidly, such as in battlefield or disaster response scenarios.

The proposed TEAHR framework represents a significant advancement in the field of WSNs, offering a robust and energy-efficient solution for trust management and routing. By integrating multiple trust metrics and focusing on energy efficiency, TEAHR addresses the critical challenges of security, reliability, and scalability in WSNs. The framework's ability to dynamically assess Trust and manage anomalies ensures that the network remains secure and operational, even in the face of evolving threats. As WSNs continue to expand and find applications in critical areas, the TEAHR framework provides a valuable tool for enhancing network performance and security. Figure 4 shows the flow of the proposed algorithm.

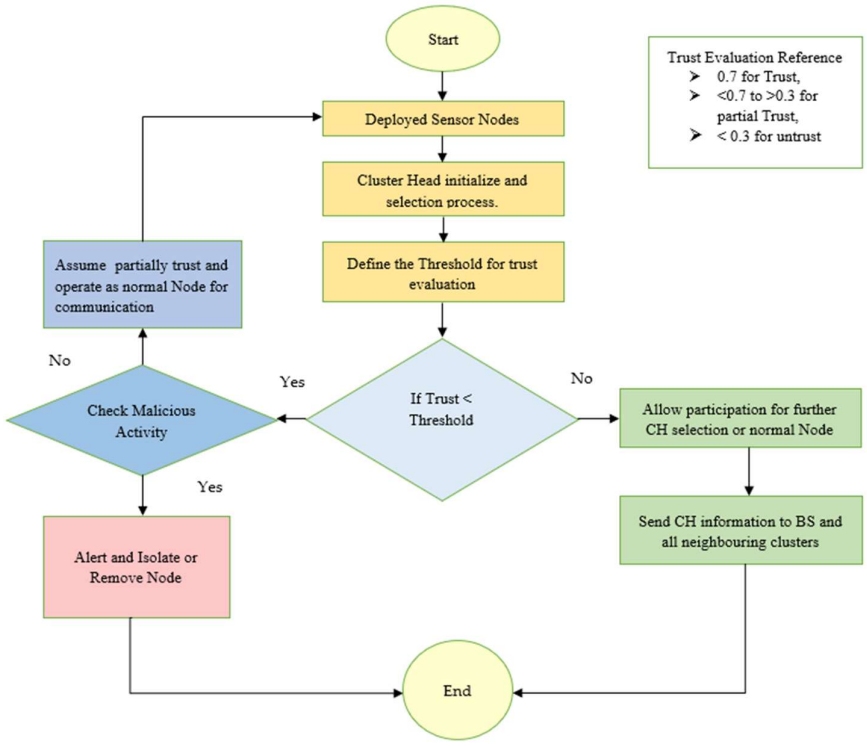

Figure 4. Flow Chart of TEAHR.

In the proposed TEAHR framework, we deploy the sensor nodes in a 1000x1000m area and start the cluster head (CH) initialization and selection process. To choose the CH, we define the threshold value (which is defined above; check the Trust Evaluation Reference) based on the energy level of each sensor node. If the trust level is less than the Threshold, then check for malicious activities. If the sensor node is a malicious (anomaly) node, then discard it; otherwise, participate in sending,

 doi:10.20944/preprints202502.1662.v1

receiving and forwarding the data to CH. If the Trust is more than the Threshold, then use those sensor nodes to select the CH for data transfer activities.

3.2. Proposed Algorithm

As per the definition of Trust, it isn't easy to create Trust in between the dynamic nature of wireless sensor networks. In our proposed work, we tried to create Trust between sensor-to-sensor nodes and sensor nodes to cluster heads, as well as between the cluster heads. So, on the basis of this Trust, we measure the anomaly in the wireless sensor network, and we propose a framework for that. In the proposed algorithm, it is designed to evaluate the trustworthiness of nodes within a WSN. The variables used in this algorithm are xi: Represents an individual standard (non-cluster head) Node within the network. CH: Represents a cluster-head node. Cluster heads are specific nodes that take on additional responsibilities, such as coordinating communication within their cluster. CH.T: This is the Trust Table associated with a Cluster Head. It stores trust-related information about the normal nodes in that cluster. On the basis of these variables, the Trust is calculated as follows: T_energy(x_i): Energy Trust. Measures how much energy a node has remaining relative to its maximum capacity (E_res(x_i) / E_max). Nodes running low on energy might be more susceptible to compromise. T_forward(x_i): Forwarding Trust. Calculates how reliably a node forwards data packets as intended (P_forwarded(x_i) / P_received(x_i)). T_consistency(x_i): Consistency Trust.

Ensures that nodes are forwarding data packets consistently (P_consistent(x_i) / P_forwarded(x_i)). Inconsistent behaviour could indicate problems. T_behavior(x_i): Behavior Trust. A more complex measure is based on whether the Node follows the network rules and its past activity. So, overall, Trust and Threshold T_overall(x_i): Combined trust score for the Node, calculated using a weighted sum of individual trust factors (with weights w1, w2, w3, w4). T_threshold: Dynamically adjusted Threshold based on the network's current state and past data. This allows the algorithm to adapt to changing conditions. The algorithm of the proposed work is as follows:

Algorithm-1: Cluster Head Selection

Input: x_i represents the normal Node, and CH represents a Cluster-Head node.

Output: Cluster head selection

```
function cluster_head_selection(N, CH)
    for each xi in CH's cluster do
        if xi in CH.T then
            T_energy(xi) ⇐ E_res(xi) / E_max
            T_forward(xi) ⇐ P_forwarded(xi) / P_received(xi)
            T_consistency(x_i) ⇐ P_consistent(x_i) / P_forwarded(x_i)
            T_behavior(xi) ⇐ evaluate_behavior(xi)
            T_overall(xi) ⇐ w1*T_energy(xi) + w2*T_forward(xi) + w3*T_consistency(xi) + w4*T_behavior(xi)
            T_threshold ⇐ set_dynamic_trust_threshold()
            if T_overall(xi) < T_threshold then
                if is_malicious(xi) then
                    broadcast_alert(xi)
                    block_node(xi)
                else
                    flag_for_re_evaluation(xi)
                    update_trust_level(CH.T, xi)
                end if
            else
                confirm_participation(xi)
                update_trust_level(CH.T, xi)
            end if
        end if
```

 doi:10.20944/preprints202502.1662.v1

23: end for
24: end function

Line 1-2 Initialization of cluster_head_selection, which takes the network N and cluster head CH. Loop through each node x_i in the cluster of CH. Line 3-9 If x_i is in the CH.T trust table, calculate the trust metrics, i.e. Energy Trust, Forwarding Trust, Consistency Trust, and Behaviour Trust. Then, compute the overall trust score T_overall(x_i)T_overall(x_i)T_overall(x_i) based on the weighted sum of the individual trust metrics. Then, a dynamic trust threshold T will be set based on network conditions and historical data. Line 10-21 will perform the decision to find the Trust if the overall trust score T_overall(x_i) is below the Threshold: If x_i shows signs of malicious behaviour, then broadcast an alert about x_i and block x_i from all network activities. Otherwise, Flag x_i for re-evaluation to limit its network participation and update the trust level of x_i in the trust table CH.T. If the overall trust score T_overall(x_i) is above the Threshold, then confirm x_i participation in network activities and update the trust level of x_i in the trust table CH.T. Line 22 End the loop for each Node in the cluster.

Algorithm-2: Setup Dynamic Trust Thresholds

Input: xi represents a normal node, and CH denotes a cluster-head node managing a cluster of nodes.

Output: Dynamic threshold value

1: function setup_dynamic_trust_thresholds(N, CH)
2: for each x_i in CH's cluster do
3: T_threshold ⇐ calculate_initial_threshold(x_i)
4: continuously_update_threshold(T_threshold)
5: end for
6: for each x_i in N do
7: if x_i in CH.T then
8: T_energy(x_i) ⇐ E_res(x_i) / E_max
9: T_forward(x_i) ⇐ P_forwarded(x_i) / P_received(x_i)
10: T_consistency(x_i) P_consistent(x_i) / P_forwarded(x_i)
11: T_behavior(x_i) ⇐ evaluate_behavior(x_i)
12: T_overall(x_i) ⇐ w1*T_energy(x_i) + w2*T_forward(x_i) + w3*T_consistency(x_i) + w4*T_behavior(x_i)
13: adjust_routes_based_on_trust(T_overall(x_i), T_threshold)
14: if T_overall(x_i) < T_threshold then
15: redirect_critical_communications(xi)
16: end if
17: end if
18: end for
19: for each new_node in N do
20: if new_node meets_or_exceeds T_threshold then
21: integrate_node(new_node)
22: else
23: if probation_needed(new_node) then
24: extend_probation(new_node)
25: else
26: remove_node(new_node)
27: end if
28: end if
29: end for
30: periodically_recalculate_thresholds()
31: for each significant_event in network_events do
32: recalculate_T_threshold()

 doi:10.20944/preprints202502.1662.v1

33: set_thresholds_using_predictive_analytics()
34: end for
35: for each x_i in N do
36: if T_overall(x_i) < T_threshold and is_malicious(x_i), then
37: isolate_and_remove(x_i)
38: broadcast_alerts(x_i)
39: quarantine_node(x_i)
40: end if
41: update_CH_T()
42: end for
43: end function

Lines 1-5 Define the function setup_dynamic_trust_thresholds, which takes the network N and cluster head CH. For each node xi in the cluster of CH, calculate the initial trust threshold T_threshold and continuously update it. Lines 6-17 evaluate the Trust for all nodes the same as in algorithm 1 with additional adjusted routes dynamically based on the latest trust assessments. If the overall trust score T_overall(x_i) is below the threshold T_threshold, redirect critical communications away from x_i. Lines 19-29: If a new member is added for each new Nodenode in network N if the new Nodenode meets or exceeds the trust threshold T_threshold, fully integrate the Nodenode into the network with standard privileges. If the new Node fails to meet the trust threshold T_threshold: If probation is needed, extend the probation period for the Node. Otherwise, the Node can be removed from the network based on security policies. Line 30-34 Periodically recalculate the trust thresholds. Recalculate T_threshold during significant network events, such as node failures or attack detection. Use adaptive algorithms to set thresholds based on predictive analytics of network conditions and node behaviour patterns. Line 35-42 manages the malicious behaviour; if the overall trust score falls below T_threshold and the Node shows signs of malicious behaviour, then isolate and remove xi from critical communication routes and broadcast alerts. Line 43 is the end of the loop.

Algorithm-3: Trust-Based Hierarchical Routing

Input: N - Set of normal nodes, CH - Set of Cluster Heads, BS - Base Station

Output: Efficient and secure routing of data based on Trust

1: function initialize_network(N, CH, BS)
2: Deploy N sensor nodes randomly in the network area
3: Divide N into M clusters using a clustering algorithm
4: Select initial CHs using cluster_head_selection(N, CH)
5: Initialize trust levels for all nodes in N and CHs in CH
6: Set up initial routing tables for each Node and CH
7: end function
8: function trust_based_routing_setup(N, CH, BS)
9: for each node x_i in N do
10: Identify the closest Cluster Head CH_i based on minimum energy cost
11: if T_overall(x_i) >= T_threshold then
12: Assign node x_i to cluster CH_i
13: Update routing table of xi to CH_i
14: else
15: Flag x_i for re-evaluation and assign to probationary CH
16: end if
17: end for
18: end function
19: function trust_based_data_transmission(N, CH, BS)
20: for each Cluster Head CH_i in CH do
21: Aggregate data from all nodes in CH_i's cluster
22: Calculate Chi _Trust based on T_overall values of nodes in the cluster

 doi:10.20944/preprints202502.1662.v1


```
        if CHi_trust >= T_threshold then
            Transmit aggregated data to BS using shortest path routing
        else
            Redirect data transmission to secondary or neighbouring CH
            Trigger cluster_head_selection(N, CH) for re-election of CH
        end if
    end for
end function
function dynamic_trust_management(N, CH, BS)
    for each node xi in N do
        Recalculate T_overall(xi) using updated parameters
        Adjust cluster membership based on T_overall(xi)
        if T_overall(xi) < T_threshold then
            Trigger re-evaluation or isolate xi based on severity
            Update routing tables accordingly
        end if
    end for
    Update dynamic trust thresholds using setup_dynamic_trust_thresholds(N, CH)
    Adjust routes based on updated trust values and thresholds
end function
function secure_data_transmission(N, CH, BS)
    for each Cluster Head CHi in CH do
        Verify T_overall(CHi) and establish a secure communication link
        Transmit data to BS via trusted CHs or multi-hop routing if required
        if link failure or malicious behaviour is detected, then
            Isolate and reroute data through alternative CH or path
        end if
    end for
    Periodically monitor the network for any malicious(anomaly) activities
end function
function maintain_network(N, CH, BS)
    for each node xi in N do
        Monitor energy levels and T_overall(xi)
                Re-elect CHs based on residual energy and trust levels using cluster_head_selection(N, CH)
    end for
    for each Cluster Head CHi in CH do
        Evaluate performance and reassign nodes if necessary
    end for
    Perform periodic network maintenance and recalibrate trust parameters
end function
```

Trust-based hierarchical routing oversees the routing process through trust assessments and energy factors. This method invokes method 1 (cluster_head_selection(N, CH)) to pick cluster heads based on Trust and employs the dynamic trust thresholds established by Algorithm 2 for routing decisions. Establishes a hierarchical routing mechanism that is energy-efficient and trust-centric, guaranteeing adequate and secure data transmission across the network. In this algorithm, trust-based routing, data transmission, dynamic trust management, and secure data transmission functions are used to create hierarchical routing.

3.2.1. Theoretical analysis of TEAHR

 doi:10.20944/preprints202502.1662.v1

In this, we show the theoretical analysis and prove it by providing a solid induction method to show how the TEAHR algorithm is effective against malicious attacks:

Theorem 1 TEAHR is robustness behaviour against malicious CMs during CM-to-CM trust estimation.

Proof by Strong Induction:

Case 1, where there is **no interaction** between CM x and CM y, which implies $Ux,y_{C,D}(\Delta t)+Sx,y_{C,D}(\Delta t)=0$ where $Ux,y_{C,D}(\Delta t) + Sx,y_{C,D}(\Delta t) = 0$, where $Ux,y_{C,D}(\Delta t)$ represents the unsuccessful interactions between CM x and CM y. $Sx,_{C,D}(\Delta t)$ represents the successful interactions between CM x and CM y. Since there are no interactions (Ux,y+Sx,y=0), the trust value $Tx,_{C,D}(\Delta t)$ between CM x and CM y is **not directly computed by the CMs** themselves. Instead, it is computed by the **Cluster Head (CH)**, which is the most **reliable** and **trustworthy** Node, based on data from other interactions. Since the CH is trusted, malicious behaviour cannot influence the trust computation. TEAHR remains robust when there is no interaction because the CH computes the trust score, and no deception occurs. Assume that TEAHR is robust and successfully prevents deception for all cases where the sum of interactions $Ux,_{C,D}(\Delta t) + Sx,y_{C,D}(\Delta t) < K$. This means that for any number of unsuccessful and successful interactions up to k, TEAHR is capable of preventing deception. Now, we need to prove that TEAHR remains robust after k+1 interactions.

Case 2: If Sx,y C, D(Δt)=0 and Ux,y C, D(Δt)≥1. In this case, there is **at least one unsuccessful interaction** between CM x and CM y, and no successful interaction has occurred. $Sx,y_{C,D}(\Delta t)=0$ and $Ux,y_{C,D}(\Delta t)\geq 1$. This implies that CM y has not managed to establish any successful interaction with CM x. Therefore, the trust value is Tx,y C, D(Δt)=0; this trust value indicates that **CM y cannot deceive CM x** since CM x does not trust CM y after failed interactions. As a result, the system is robust against malicious behaviour in this scenario, and CM y cannot deceive CM x.

Case 3: If $Ux,y_{C,D}(\Delta t) + Sx,y_{C,D}(\Delta t) \geq 1$ and $Ux,y_{C,D}(\Delta t) \geq Sx,y_{C,D}(\Delta t)$, in this case, there is at least one interaction, and the number of unsuccessful interactions is greater than or equal to the number of successful interactions. $Ux,y_{C,D}(\Delta t) \geq Sx,y_{C,D}(\Delta t)$, this means that CM y has exhibited more or equal unsuccessful behaviours compared to successful interactions with CM x. $Tx,y_{C,D}(\Delta T) = \frac{Sx,y_{C,D}(\Delta t)}{Sx,y_{C,D}(\Delta t) + Ux,y_{C,D}(\Delta t)}$, since $Ux,y_{C,D}(\Delta t) \geq Sx,y_{C,D}(\Delta t)$. $\frac{Sx,y_{C,D}(\Delta t)}{Sx,y_{C,D}(\Delta t) + Ux,y_{C,D}(\Delta t)} < 0.5$, thus the trust value $Tx,y_{C,D}(\Delta t) < \Theta$ for any value of η and ϕ. Since the trust value falls below the Threshold Θ, this contradicts the possibility that CM y can deceive CM x, because TEAHR flags CM y as malicious or untrustworthy. TEAHR remains robust because CM yyy's malicious behaviour leads to a trust score that is lower than the Threshold, and TEAHR prevents deception.

Theorem 2 TEAHR is robust against malicious behaviour of CHs during CH-to-CH trust estimation.

Proof by Strong Induction:

Assume that TEAHR is robust for all cases where the number of interactions between CH x and CH y is n=1 to k, where the system correctly prevents deception up to k interactions.

Inductive Step: We need to prove that TEAHR remains robust after k+1 interactions between CH x and CH y.

Case 1: If $Sx,y_{CH}(\Delta t)=0$ and $Ux,y_{CH}(\Delta t) \geq 1$. In this case, there is **at least one unsuccessful interaction** between CH xxx and CH yyy, and no successful interaction has occurred. **$Sx,y_{CH}(\Delta t)=0$ and $Ux,y_{CH}(\Delta t) \geq 1$.** This implies that CH y has failed to establish a successful interaction with CH x. Therefore, the trust value between CH x and CH y is $Tx,y_{CH}(\Delta t) = 0$; this means that **CH y cannot deceive CH x** since the system assigns a zero trust value due to the absence of successful interactions. The robustness of TEAHR ensures that malicious Cluster Heads cannot deceive others by simply interacting unsuccessfully. So, TEAHR remains robust when there are only unsuccessful interactions between CHs because the trust score is set to zero, thus preventing deception.

Case 2: If $Ux,y_{CH}(\Delta t) + Sx,y_{CH}(\Delta t) \geq 1$, in this case, there is **at least one interaction** between CH x and CH y, and the number of unsuccessful interactions is greater than or equal to the number of successful interactions. $Ux,y_{CH}(\Delta t) \geq Sx,y_{CH}(\Delta t)$. The trust score $Tx,y_{CH}(\Delta t)$ between CH x and CH y is computed as:

 doi:10.20944/preprints202502.1662.v1

$T_{x,y_{CH}}(\Delta t) = \frac{S_{x,y_{CH}}(\Delta t)}{S_{x,y_{CH}}(\Delta t) + U_{x,y_{CH}}(\Delta t)}$, since $U_{x,y_{CH}}(\Delta t) \geq S_{x,y_{CH}}(\Delta t)$ the term $\frac{S_{x,y_{CH}}(\Delta t)}{S_{x,y_{CH}}(\Delta t) + U_{x,y_{CH}}(\Delta t)} < 0.5$. Thus, the trust value $T_{x,y_{CH}}(\Delta t) < \Theta$, which means that TEAHR flags CH y as malicious if the trust score falls below the malicious Threshold Θ. This contradiction to the hypothesis that CH y could deceive CH x ensures that TEAHR prevents any attack or deception. TEAHR remains robust as the trust score correctly falls below the malicious Threshold, flagging CH y and preventing deception.

Figure 5 shows the sensor-to-sensor, sensor-to-cluster head, cluster-head-to-group head, and cluster-head-to-base station trust and communication. The critical element of a TEAHR protocol is Cluster Formation: The nodes are grouped into clusters, often based on location, energy levels, or signal strength. The head election takes energy and trust metrics. Trust Evaluation: Nodes monitor each other's behaviour (packet forwarding, data consistency, etc.). Trust models calculate trust scores, aiding in identifying unreliable nodes. Energy-Aware Routing: Routing paths are chosen to prioritize nodes with higher energy reserves. Minimize the number of hops data takes to conserve energy. Load balancing techniques prevent overtaxing of specific nodes. Secure Communication: Lightweight encryption or authentication may be used to protect data integrity and confidentiality, especially when Trust is involved.

 doi:10.20944/preprints202502.1662.v1

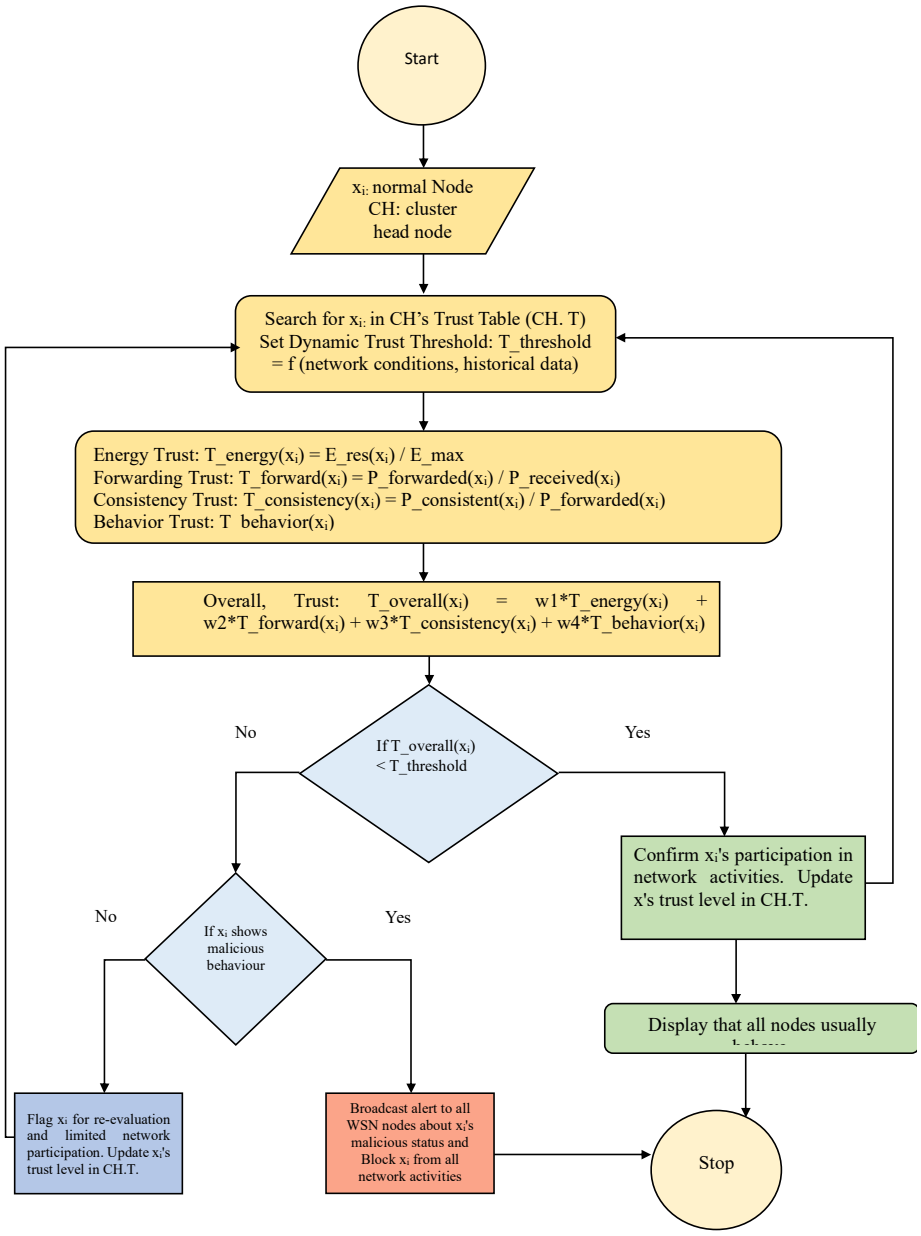

Figure 5. Flow of CH Selection.

Result and Discussion

2.1. Trusted Energy Aware Hierarchical Routing (TEAHR)

Authors[31] demonstrate the effectiveness of the TIOCHR algorithm in enhancing the network lifetime and energy efficiency; it needs to fully explore the potential challenges of implementing such trust-based systems in highly dynamic or adversarial environments where node reliability might frequently change. Furthermore, the study might benefit from a comparative analysis with other trust-based models to establish the specific advantages or limitations of the TIOCHR approach in various real-world applications. In our proposed model, we tried to overcome the drawback of TIOCHR.

 doi:10.20944/preprints202502.1662.v1

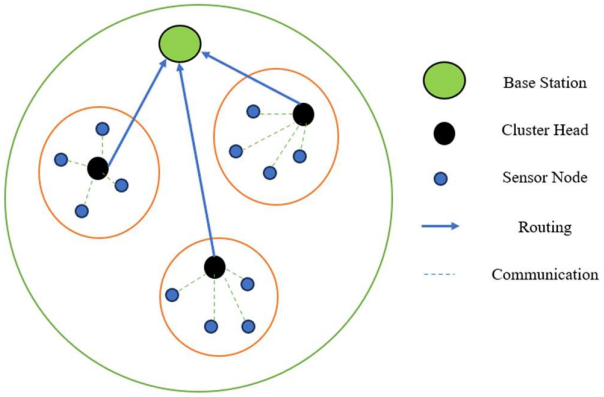

Figure 6. sensor-to-sensor, sensor-to-group head, cluster head to cluster head-to-group head and cluster head to base station trust and routing.

Table 2. Notation Description.

Notation	Description
$T_{basic}(\mathrm{i}),\ E_{res}(\mathrm{i}), E_{ref}$	The fundamental trust score of Node I is based on its residual energy relative to a reference energy level.
$T_{norm}(\mathrm{i}), E_{res}(\mathrm{i}), E_{max}$	Normalized trust score of nodes i relative to the maximum residual energy in the network.
$T_{weighted}(\mathrm{i}), E_{res}(\mathrm{i}), E_{ref}, w_1, w_2, \mathrm{F(i)}$	The weighted trust score of nodes considers energy-based Trust and additional factors like centrality or mobility.
$P'_{ch}(\mathrm{j}), P_{ch}(\mathrm{j}), E_{res}(\mathrm{j}), E_{avg\_ann} - 1$	Updated CH probability for node j based on residual energy and the average residual energy of neighbouring nodes.
$S(i,k), \beta_1, \beta_2, \beta_3, E_{res}(i), D(i,k), P_{ch}(i)$	Suitability metric for selecting a tentative CH based on residual energy, distance, and CH probability.
$M(i,j), \gamma 1, \gamma 2, \gamma 3\ \mathrm{Eres}(j), D(i,j), C(i,j)$	A composite metric for selecting the most suitable CH based on energy, distance, and communication cost.
$T_{hreshold}$, network conditions, historical data	Dynamic trust threshold that adapts based on network conditions and historical data.
$T_{new}(\mathrm{i,t}),\ T_{initial}, \delta,\ T_{observed\ actions}(\mathrm{i,t})$	The new trust value for Node i at time t was adjusted based on observed actions and initial Trust.

2.1. Initialization

Each Node in the network starts with an initial percentage of being a cluster head (CH) candidate, called the "CH probability" (P_{ch}), and sets its status to undecided. The initial P_{ch} can be a system parameter based on the size and density. Let $P_{ch}(i)$ be the probability of Node i becoming a CH. Each Node I broadcasts its $P_{ch}(i)$ along with its residual energy $E_{res}(i)$. The nodes assess their residual energy relative to a reference, which can be the initial energy or an average energy level. This step ensures that nodes with higher energy have a higher chance of becoming a cluster head. Let $E_{res}(i)$ be the residual energy of Node i, and E_{ref} be the reference energy level, which could be the initial energy level E_{init} or the average energy level E_{avg} of all the nodes in the network. The fundamental trust score T_{basic} (i) of Node i can be estimated as follows based on the residual energy of Node i:

 doi:10.20944/preprints202502.1662.v1

$$T_{basic}(\mathrm{i}) = E_{res}(\mathrm{i}) / E_{ref} \quad (16)$$

This equation means that the trust score is directly proportional to the lasting energy of a node compared with the chosen reference energy level. In a more dynamic scenario, where it becomes relevant that a chosen maximum residual energy, E_{max}, for the network at the time of evaluation could change over time, the normalized trust score would be:

$$T_{norm}(\mathrm{i}) = E_{res}(\mathrm{i}) / E_{max} \quad (17)$$

The normalization of the trust score is such that it falls within a range [0, 1], with one corresponding to a node with maximum residual energy in the network. Furthermore, F(i) can have several other factors related to the centrality of the Node, its mobility, or historical performance in trust assessment. Then, a weighted trust score $T_{weighted}(\mathrm{i})$ can be in the form:

$$T_{weighted}(\mathrm{i}) = w_1(E_{res}(\mathrm{i}) / E_{ref}) + w_2\, \mathrm{F(i)} \quad (18)$$

Where w_1 and w_2 are the weights assigned to the energy-based trust component and the additional factors, respectively, with $w_1 + w_2 = 1$. For Adaptive CH Probability Update each non-CH candidate node j, the updated CH probability P'_{ch} (j) is calculated as:

$$P'_{ch}(\mathrm{j}) = P_{ch}(\mathrm{j}) + \alpha(E_{res}(\mathrm{j})/ E_{avg\_ann} - 1) \quad (19)$$

Here, E_{avg_ann} is the average of the residual energies announced by neighbouring nodes, and α is a scaling factor to adjust the influence of the residual energy difference. So on the basis of that, the tentative CH selection Evaluation for each non-CH node k evaluates tentative CHs based on a suitability metric S(i,k), which could consider the announced Eres and Pch, and selects the one with the highest S(i,k).

$$S(i,k) = \beta_1 E_{res}(i) + \beta_2 D(i,k) + \beta_3 P_{ch}(i) \quad (20)$$

Here, D(i,k) represents the distance or signal strength between nodes i and k, and $\beta_1, \beta_2, and\ \beta_3$ are weights. Nodes with similar E_{res} might be further evaluated based on secondary parameters like proximity or communication cost C(i,k), selecting the CH that minimizes this cost. Minimise C(i,k). The final CH set is determined by the nodes that receive confirmations from at least one other Node. Each non-CH Node joins the nearest or most suitable CH based on a composite metric M(i,j) considering Eres, D(i,j), and C(i,j).

$$M(i,j) = \gamma 1\ \mathrm{Eres}(j) - \gamma 2 D(i,j) - \gamma 3 C(i,j) \quad (21)$$

Here, $\gamma 1$ $\gamma 2$ $\gamma 3$ are weights. The CH should continuously evaluate the trust values of the existing cluster members based on energy backup, packet forwarding, packet consistency, and any other relevant parameters. The network should define dynamic trust thresholds that can be adjusted over time based on network conditions and requirements. The initial trust $T_{threshold}$ can be set on historical data or network specifications.

$$T_{threshold} = f(\text{network conditions, historical data}) \quad (22)$$

This $T_{hreshold}$ is adaptive, based on the network's historical data and current conditions. Trust-based communication and packet forwarding routes within the cluster should be established based on trust values. Higher trust nodes should be preferred for critical roles, such as data aggregation or relay.

2.1. Integrating New Cluster Members

When a new node is added to the cluster, an initial trust assessment is conducted. New nodes start with an initial trust value initial based on standard network entry criteria, possibly lower than the trust threshold for existing members to account for the lack of historical behaviour data. This

 doi:10.20944/preprints202502.1662.v1

could involve directly monitoring the Node's behaviour in terms of its packet forwarding rate, energy usage, and consistency of data provided.

$$T_{new}(i,t) = T_{initial} + \delta(T_{observed\ actions}(i,t)) \quad (23)$$

Here, $\delta(T_{observed\ actions}(i,t))$ is the trust change with time t from when a new node joined based on its types of observation, such as packet forwarding behaviour, energy consumption and proper implementation of network protocols. As a first step, the initial trust threshold of new nodes can be set to slightly less than that for current members so they have time to put their performance where their mouth is. Nevertheless, this maximal Threshold should be enough to hinder obviously nefarious nodes from perturbing mission-critical roles. New nodes may serve a probationary term during which they are observed more carefully. The trust value of those peers is prone to change more often, and their network roles are limited to non-critical tasks until a steady-state level of high Trust has been achieved and security checks are performed to verify if the new Node belongs did not arise from known attacks. This could mean checking the Node's credentials, confirming that its software has not been tampered with and examining whether it is communicating in an abnormal way. Other nodes in the network watch how new ones on board participate and gradually increase their Trust as it becomes clear they are providing a benefit to the operations of that network. As the trust level of a new node grows and exceeds the dynamic Threshold, it can then be assigned additional tasks (i.e. data aggregation or being a secondary relay) within the cluster.

In our proposed algorithm, hierarchical routing is used and focuses on selecting CH from the group of sensor nodes, which is responsible for maintaining the information about the nodes within its cluster, calculating the Trust and making decisions about whether the Node is trustworthy or malicious. Our proposed algorithm, TEAHR, calculates the trustworthiness of nodes based on energy, forwarding behaviour and consistency and dynamically adjusts the Threshold to decide whether a node can participate in selecting CH or should be blocked. As mentioned, the proposed algorithm updates the trust threshold dynamically and recalculates it based on various network events, further emphasizing the hierarchical decision-making process at the CH level.

2.1. Simulation Environment

Table 2 defines the parameters used in this proposed model; in this table, we define the trustworthiness, packet forward rate, anomaly factor, energy, trust level, threshold for detecting the malicious Node (anomaly), etc.

Table 2. Parameters for calculating the Trust.

S.No.	Parameter	Description	Typical Values or Ranges
1	Trustworthiness (TB(x_i))	Range from 0 (untrustworthy) to 1 (fully trustworthy), with thresholds for roles based on value.	> 0.7 for Trust, <0.7 to >0.3 for partial Trust, < 0.3 for untrust
2	Packet Forwarding Rate (PFR)	Part of $Px_iF()$ indicates reliability in forwarding packets; closer to 1 is better.	Close to 1
3	Anomaly Factor (AF)	It reflects a deviation from expected behaviour, part of AF(x_i); being closer to 0 indicates normal behaviour.	Close to 0
4	Residual Energy for CH Selection	Nodes with more than a specified percentage of their initial energy (e.g., > 50%) are eligible for CH selection.	> 50%

 doi:10.20944/preprints202502.1662.v1

5	Trust Level for CH Selection	Minimum trust level (e.g., > 0.7) required for a node to be considered for CH selection.	> 0.7
6	Malicious Threshold (TB(x_i))	Nodes with a trust level below a certain threshold (e.g., < 0.3) are considered malicious and blocked.	< 0.3
7	Key Refresh Interval	The frequency of cryptographic key refreshes depends on security requirements (e.g., every 24 hours).	Every 24 hours or as needed
8	Trust Evaluation Interval	Frequency of trust evaluations, adjusted based on environment stability (e.g., every hour in stable environments).	Every hour to every 10 minutes
9	New Node Trust Initialization	A neutral initial trust level is assigned to new nodes until enough behavioural data is collected (e.g., 0.5).	0.5
10	Cluster Size Range	Adjustable range based on network density and area, typically 5 to 20 nodes per CH.	5 to 20 nodes per CH

2.1. Deployment of sensor nodes

In this paper, we set up the environment using 1000 x 1000 m2, and 500 sensors are randomly distributed. Figure 7 shows the deployment of the sensor nodes. Trustworthiness (TB(x_i)), Packet Forwarding Rate (PFR), Anomaly Factor (AF), Malicious Threshold (TB(x_i)), Trust Evaluation Interval, and New Node Trust z are used to evaluate the proposed work. The initial energy for each sensor node is 100J.

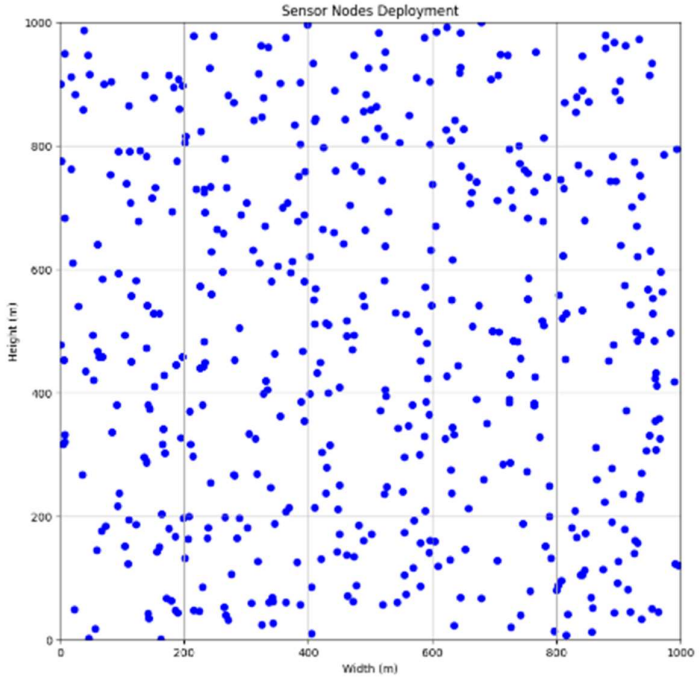

Figure 7. Sensor node deployments.

Cluster Head (CH) Selection: To select the cluster head (CH), the minimum threshold range is from 0 (untrustworthy) to 1 (fully trustworthy). The minimum trust level required for CH selection = 0.7; nodes with >50% of initial energy are eligible for CH selection=0.5. Malicious Trust

 doi:10.20944/preprints202502.1662.v1

threshold=0.3 will be considered as malicious Node, and the trust level required value is 0.7 for CH selection.

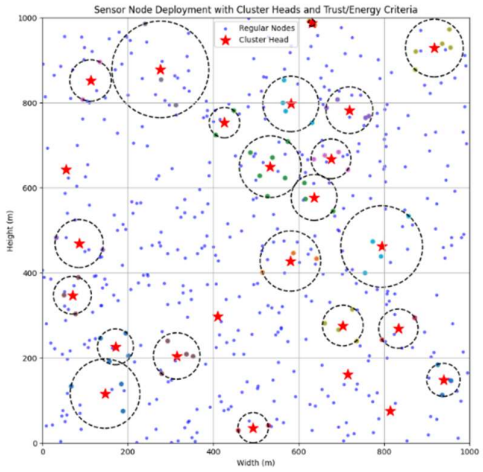

Figure 8. Cluster Head Selection.

Figure 8 displays the cluster heads chosen out of 500 sensor nodes on the basis of defined energy. Now, we insert the 20% malicious nodes to find the anomaly in the given sensor network as we insert the 20%, i.e. 100 malicious sensor nodes—the energy of overall sensor nodes after selecting the cluster head and identifying the malicious Node.

Total remaining energy in the network: 40750.00 units

Average remaining energy per Node: 81.50 units

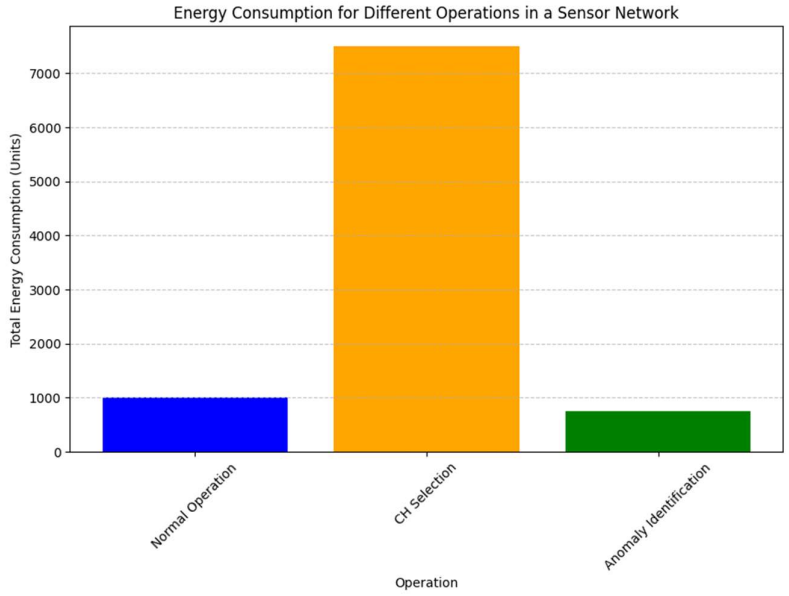

Figure 9. Energy consumption during normal operation, CH selection, and anomaly identification in WSN.

Initial energy for each node=1.0, energy consumed for forwarding a packet=0.01, Threshold below which nodes are blacklisted=0.3, number of packets each Node attempts to send=100. Figure 9 displays the energy consumption during different operations, based on the above assumptions: the successful deliveries=30000, total latency=154428, total packets sent = 50000, packet forwarding rate (pfr) = 0.616, latency (lt) = 5.013, energy consumption (ec) = 308.00, network longevity as a fraction of alive nodes(NL)=0.384, Routing Overhead(ROH)=0.0033

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 February 2025 doi:10.20944/preprints202502.1662.v1

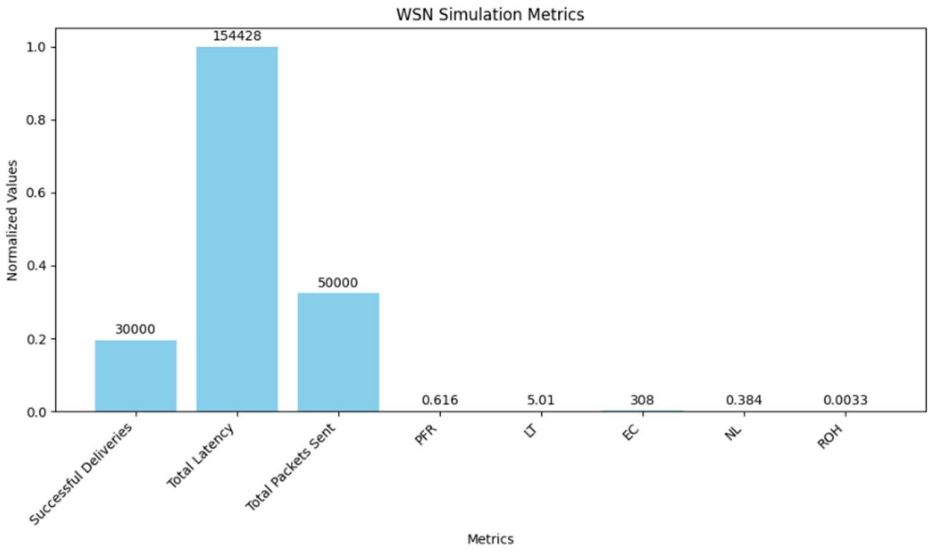

Figure 10. successful deliveries, total latency, total packets sent, pfr, lt, ec, nl, roh.

Identifying the suitable algorithm requires a balance between energy efficiency and latency, especially as the network grows. The graph below represents the trade-offs that different algorithms present when scaling up a network. TIOCHR and especially TEAHR appear to be more scalable solutions with respect to latency.

2.1. Analysis of TEAHR

In Figure 11, the proposed algorithm is compared with other algorithms, and the proposed algorithm reduces total latency by 15% after inserting the anomaly as defined above. As shown in Figure 10, the average latency of the TEAHR algorithm is more stable than other algorithms.

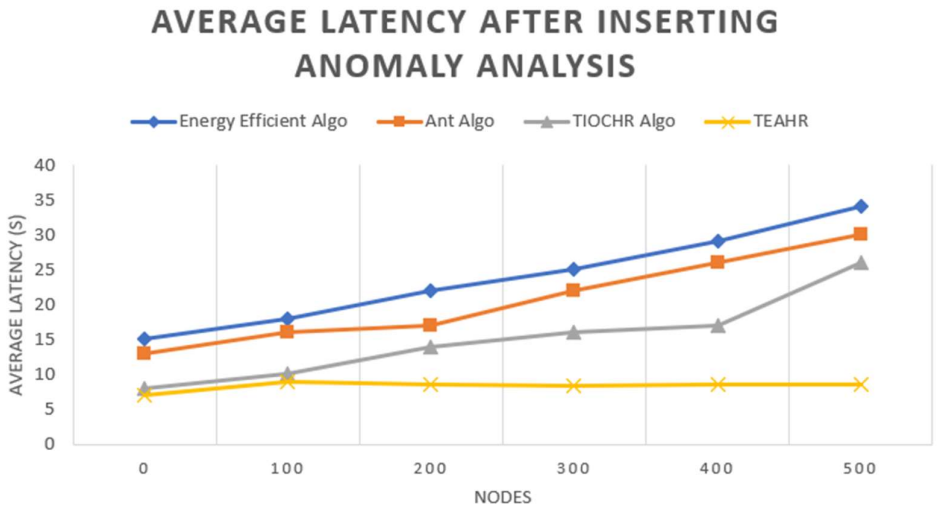

Figure 11. Average latency analysis Energy Efficient Algo[32], Ant Algo[33], TIOCHR Al.go[31]

In Figure 12, we represent the packet forwarding rate analysis when the network grows; in this figure, all four algorithms start with high efficiency in smaller networks but begin to diverge as more nodes are introduced. The

 doi:10.20944/preprints202502.1662.v1

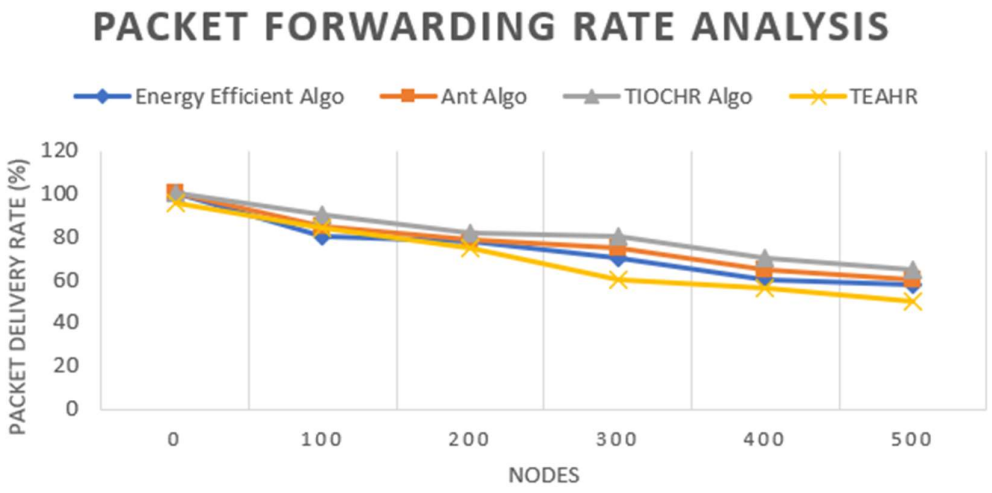

Figure 12. Packet forward rate analysis Energy Efficient Algo[32], Ant Algo[33], TIOCHR Al.go[31]

Energy Efficient and Ant algorithms show a moderate decrease in packet delivery rate, which might be acceptable in many applications. The TIOCHR algorithm performs slightly better, implying a more robust handling of packet forwarding under scaling conditions. The TEAHR algorithm stands out for its remarkable stability, indicating a robust approach to packet forwarding that is less affected by network scaling.

Energy consumption plays a pivotal role over time in wireless sensor networks; in Figure 13, all algorithms increase their energy consumption over time; none of them shows a decrease in energy, which might be expected as resources are consumed and tasks become more energy-intensive. The TEAHR algorithm stands out as the most energy-conserving algorithm throughout the simulation, maintains a stable packet forwarding rate and enhances energy efficiency by around 20%. This characteristic could make it a preferred choice in scenarios where energy conservation is crucial, such as in battery-powered devices or remote sensing equipment. The initial energy efficiency of the TIOCHR algorithm may be advantageous in short-term applications, but for more extended simulations or real-time applications, its energy usage becomes less favourable.

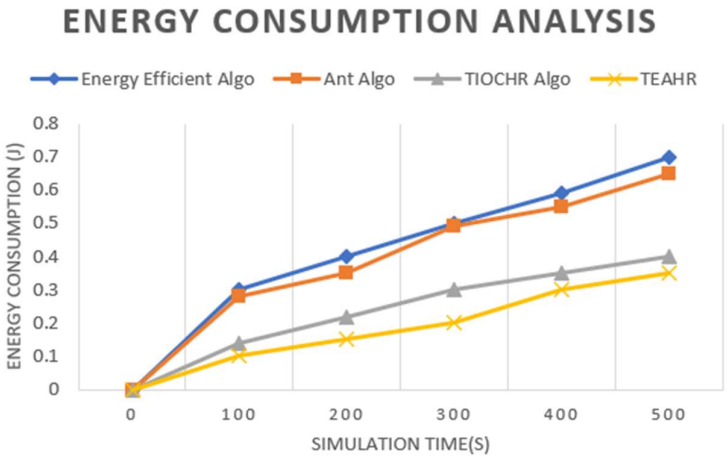

Figure 13. Energy Consumption Analysis Energy Efficient Algo[32], Ant Algo[33], TIOCHR Al.go[31]

Conclusion

By evaluating the consistency of sensor nodes based on their behaviour and performance, trust management is a multi-pronged method that improves the safety of wireless sensor networks

 doi:10.20944/preprints202502.1662.v1

(WSNs). This is accomplished by analyzing the consistency of sensor nodes. Through the utilization of mathematical models and trust evaluation techniques, wireless sensor networks (WSNs) are able to design communication channels that are safe and protect themselves against threats that originate from within the network boundaries. As the state of the art in technology continues to improve, there is little question that trust management systems will continue to play an essential role in guaranteeing the security and efficiency of wireless sensor networks. Our Trusted Energy-Aware Hierarchical Routing (TEAHR) for Wireless Sensor Networks displays good performance across a variety of parameters, such as an examination of the average latency, an analysis of the packet forwarding rate, and an analysis of the energy usage. After doing these evaluations, it was discovered that the TEAHR algorithm had the lowest energy usage throughout the duration of the simulation. As a result, it was determined to be the most energy-efficient algorithm among those that were selected for comparison. TEAHR dynamic method makes this algorithm more robust; we need to further test this by utilizing a hybrid technique; then, we can make sure that the algorithm is more resilient, energy efficient, and secure enough to detect anomalies in wireless sensor networks by maintaining Trust among the sensor's nodes. To do this, we need to make sure that the algorithm is able to detect anomalies in wireless sensor networks. Our proposed algorithm reduces total latency by 15%, enhances energy efficiency by around 20%, and maintains a stable packet forwarding rate. This algorithm is even more potent because of the TEAHR dynamic approach.

 doi:10.20944/preprints202502.1662.v1

Author Contributions Vikas[1]: Writing – original draft, Methodology. Charu Wahi[2]: Writing – Supervision, Conceptualization, Formal analysis. B B Sagar[3]: Writing – Supervision, Conceptualization, Formal analysis. Manisha Manjul[4]: Writing, Conceptualization, Formal analysis, Investigation, review, and editing.

Data Availability Statement: The datasets generated and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare that they have no competing interests.

Funding Declaration: We declare that this manuscript received no funding from any sources.

References

1. Ye, Z., Wen, T., Liu, Z., Song, X. & Fu, C. An Efficient Dynamic Trust Evaluation Model for Wireless Sensor Networks. J Sens 2017, 1–16 (2017).
2. Fang, W. et al. Trust-Based Attack and Defense in Wireless Sensor Networks: A Survey. Wirel Commun Mob Comput 2020, 1–20 (2020).
3. Chen, Z., Tian, L. & Lin, C. Trust Model of Wireless Sensor Networks and Its Application in Data Fusion. Sensors 17, 703 (2017).
4. Vikas, Sagar, B. B. & Munjul, M. Security issues in wireless sensor network – A survey. Journal of Discrete Mathematical Sciences and Cryptography 24, 1415–1427 (2021).
5. Fang, W., Cui, N., Chen, W., Zhang, W. & Chen, Y. A Trust-Based Security System for Data Collection in Smart City. IEEE Trans Industr Inform 17, 4131–4140 (2021).
6. Djedjig, N., Tandjaoui, D., Romdhani, I. & Medjek, F. Trust Management in the Internet of Things. in 122–146 (2018). doi:10.4018/978-1-5225-5736-4.ch007.
7. Mon, S. F. A., Winster, S. G. & Ramesh, R. Trust Model for IoT Using Cluster Analysis: A Centralized Approach. Wirel Pers Commun 127, 715–736 (2022).
8. Prathap, U., Shenoy, P. D. & Venugopal, K. R. PCAD: Power control attack detection in wireless sensor networks. in 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) 1–6 (IEEE, 2016). doi:10.1109/ANTS.2016.7947798.
9. Chae, Y., DiPippo, L. C. & Sun, Y. L. Trust Management for Defending On-Off Attacks. IEEE Transactions on Parallel and Distributed Systems 26, 1178–1191 (2015).
10. Wagh, S., More, A. & Khavnekar, A. Identification of selfish attack in cognitive radio ad-hoc networks. in 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) 1–4 (IEEE, 2015). doi:10.1109/ICCIC.2015.7435805.
11. Dong, W. & Liu, X. Robust and Secure Time-Synchronization Against Sybil Attacks for Sensor Networks. IEEE Trans Industr Inform 11, 1482–1491 (2015).
12. Ajaykumar, N., Sarvagya, M. & Parandkar, P. A novel security algorithm ECC-L for wireless sensor network. Internet Technology Letters 3, (2020).
13. Fang, W., Zhang, W., Yang, Y., Liu, Y. & Chen, W. A resilient trust management scheme for defending against reputation time-varying attacks based on BETA distribution. Science China Information Sciences 60, 040305 (2017).
14. Feng, R., Han, X., Liu, Q. & Yu, N. A Credible Bayesian-Based Trust Management Scheme for Wireless Sensor Networks. Int J Distrib Sens Netw 11, 678926 (2015).
15. Wahi, C., Chakraverty, S. & Bhattacherjee, V. A trust-based secure AODV routing scheme for MANET. International Journal of Ad Hoc and Ubiquitous Computing 38, 231 (2021).
16. Alshehri, M. D. & Hussain, F. K. A fuzzy security protocol for trust management in the internet of things (Fuzzy-IoT). Computing 101, 791–818 (2019).
17. Busi Reddy, V., Venkataraman, S. & Negi, A. Communication and Data Trust for Wireless Sensor Networks Using D–S Theory. IEEE Sens J 17, 3921–3929 (2017).
18. Ram Prabha, V. & Latha, P. Fuzzy Trust Protocol for Malicious Node Detection in Wireless Sensor Networks. Wirel Pers Commun 94, 2549–2559 (2017).

 doi:10.20944/preprints202502.1662.v1

19. Patra, M. & Acharya, S. A Fuzzy Logic-Based Trust Management Scheme for Wireless Sensor Network. in 155–166 (2022). doi:10.1007/978-981-19-1018-0_14.
20. Han, D., Du, X., Wang, L., Liu, X. & Tian, X. Trust-Aware and Fuzzy Logic-Based Reliable Layering Routing Protocol for Underwater Acoustic Networks. Sensors 23, 9323 (2023).
21. Patel, R. & Acharya, S. A Fuzzy Logic Based Trust Evaluation Model for IoT. in 147–156 (2024). doi:10.1007/978-981-99-3932-9_14.
22. Abdalzaher, M. et al. Game Theory Meets Wireless Sensor Networks Security Requirements and Threats Mitigation: A Survey. Sensors 16, 1003 (2016).
23. Abdalzaher, M. S., Samy, L. & Muta, O. Non-zero-sum game-based trust model to enhance wireless sensor networks security for IoT applications. IET Wireless Sensor Systems 9, 218–226 (2019).
24. Adnan, M., Yang, T., Das, S. K. & Ahmad, T. Utility of Game Theory in Defensive Wireless Sensor Networks (WSNs). in 895–904 (2021). doi:10.1007/978-981-15-5113-0_75.
25. Wang, E. K., Chen, C.-M., Zhao, D., Ip, W. H. & Yung, K. L. A dynamic trust model in internet of things. Soft comput 24, 5773–5782 (2020).
26. Yu, X. et al. Trust-based secure directed diffusion routing protocol in WSN. J Ambient Intell Humaniz Comput 13, 1405–1417 (2022).
27. Parvathi, C. & Talanki, S. Energy Saving Hierarchical Routing Protocol in WSN. in Wireless Sensor Networks - Design, Deployment and Applications (IntechOpen, 2021). doi:10.5772/intechopen.93595.
28. Palan, N. G., Barbadekar, B. V. & Patil, S. Low energy adaptive clustering hierarchy (LEACH) protocol: A retrospective analysis. in 2017 International Conference on Inventive Systems and Control (ICISC) 1–12 (IEEE, 2017). doi:10.1109/ICISC.2017.8068715.
29. Lindsey, S. & Raghavendra, C. S. PEGASIS: Power-efficient gathering in sensor information systems. in Proceedings, IEEE Aerospace Conference 3-1125-3–1130 (IEEE). doi:10.1109/AERO.2002.1035242.
30. Perkins, C. E. & Royer, E. M. Ad-hoc on-demand distance vector routing. in Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications 90–100 (IEEE, 1999). doi:10.1109/MCSA.1999.749281.
31. Babu, S. S. & Geethanjali, N. Lifetime improvement of wireless sensor networks by employing Trust Index Optimized Cluster Head Routing (TIOCHR). Measurement: Sensors 32, 101068 (2024).
32. Mostafaei, H. Energy-Efficient Algorithm for Reliable Routing of Wireless Sensor Networks. IEEE Transactions on Industrial Electronics 66, 5567–5575 (2019).
33. Li, X., Keegan, B., Mtenzi, F., Weise, T. & Tan, M. Energy-Efficient Load Balancing Ant Based Routing Algorithm for Wireless Sensor Networks. IEEE Access 7, 113182–113196 (2019).