applied sciences

MDPI

Review

Trust Models in Wireless Sensor Networks for Defending Against Denial-of-Service Attacks: A Literature Review

Lijuan Wang, Krassie Petrova * and Mee Loong Yang

School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand; lijuan.wang@autuni.ac.nz (L.W.); bobby.yang@aut.ac.nz (M.L.Y.)
* Correspondence: krassie.petrova@aut.ac.nz

Abstract: Denial-of-service (DoS) attacks pose a significant threat to wireless sensor networks (WSNs) and are an impediment to their mass deployment. Current research has identified trust models as a plausible defense against DoS attacks. However, most of the proposed solutions focus only on one or two specific DoS attack scenarios and provide very limited guidance to a WSN with regard to important practical aspects, such as setting up threshold configurations and weight allocation schemes. This study conducts a comparative analysis of relevant work to build a foundation for the development of robust trust models that can counter a range of different types of DoS attacks in WSNs. In particular, this study examines the required trust evidence, the methods for extracting trust evidence, and the trust evaluation techniques for developing effective trust models. This study identifies the challenges in the implementation of trust models, such as the need to determine feasible threshold limits and trust metric weightings and the need to manage the loss of trust information. In addition, this study explores link quality and node authentication as factors affecting trust evaluation and the integration of trust models with network routing protocols.

Keywords: wireless sensor networks; WSNs; trust models; trust evidence; trust evaluation; denial-of-service attacks; DoS attacks

Academic Editor: Juan Antonio López Ramos

Received: 19 December 2024
Revised: 7 March 2025
Accepted: 10 March 2025
Published: 12 March 2025

Citation: Wang, L.; Petrova, K.; Yang, M.L. Trust Models in Wireless Sensor Networks for Defending Against Denial-of-Service Attacks: A Literature Review. *Appl. Sci.* **2025**, *15*, 3075. https://doi.org/10.3390/app15063075

1. Introduction

1.1. Wireless Sensor Networks and Network Architecture Types

Wireless sensor networks (WSNs) are used widely as a means of monitoring and collecting data from the environment they are deployed in. The small, battery-powered network nodes are capable of sensing, processing, and transmitting data using wireless communication protocols [1]. Normally, the data collected by individual nodes are sent to a sink node or a base station connected to the WSN. From there, the data are transmitted through the Internet to a database (DB) server for further processing and analysis. In general, the communication architecture of a WSN can be flat or hierarchical [2,3]. As shown in Figure 1, in a flat WSN, the networks nodes transmit data to the sink node via multi-hop paths. In a hierarchical (cluster-based) WSN, the network nodes are grouped into clusters; the nodes send data to the cluster head (CH), which then relays the data to the sink node (Figure 2). In addition, a WSN can deploy a software-defined network (SDN) architecture in which the network control plane is separated from the data plane (Figure 3). A software-defined wireless network (SDWSN) supports network scalability and node recycling [4].

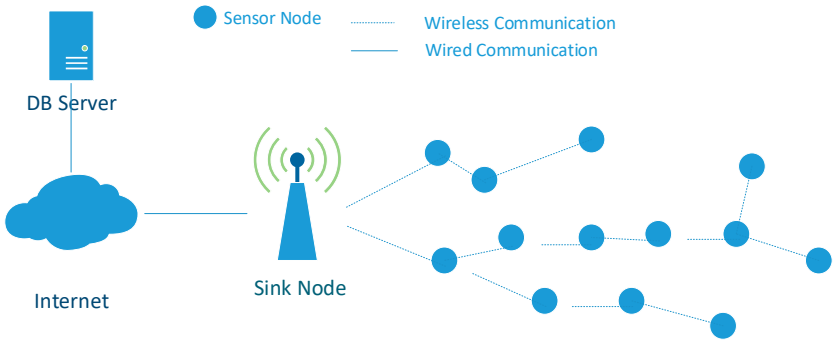

Figure 1. The network architecture of a flat WSN.

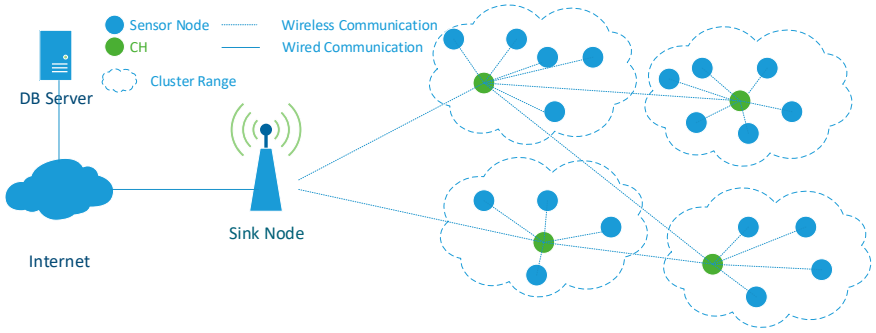

Figure 2. The network architecture of a cluster-based WSN.

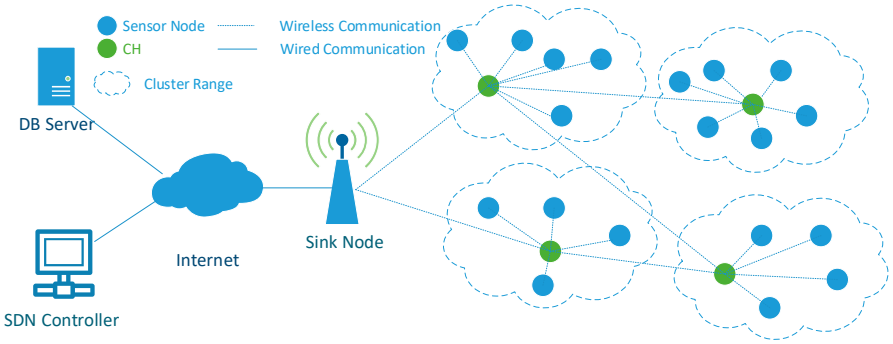

Figure 3. The network architecture of an SDWSN.

1.2. WSN Vulnerabilities and Trust Models

WSNs are commonly deployed to support Internet of Things (IoT) services and capabilities. However, WSNs are vulnerable to node failure, to internal attacks, such as tampering with WSB nodes, and to Internet-based denial-of-service (DoS) attacks. This poses a serious risk to systems relying on data collected by WSNs.

For instance, a faulty leaf wetness sensor's reading in a humidity lab remained stuck at the maximum value of the measurement range for several days. As a consequence, the temperature and humidity sensors housed in the same box in the lab short-circuited, causing erroneous temperature readings [5]. The high failure rate of sensors deployed in harsh environments means that faulty nodes can produce incorrect readings, potentially leading to errors in the overall measurement results.

Low-cost, non-tamper-resistant nodes can be captured by a malicious actor, who can then exploit them to obtain access to the cryptographic keys used and launch an internal attack from the compromised node. Furthermore, WSN nodes can be compromised by an externally launched DoS attack.

According to Wood et al. [6], a DoS attack is an event that diminishes or eliminates a network's capacity to perform its expected function. This is especially important in critical

applications, for example, a WSN that gathers and transmits signals from medical implants. In this case, a DoS attack can compromise the availability of the data and threaten the well-being of a human patient [7].

Some typical DoS attacks on sensor networks are described in [6]. While most of them are similar to DoS attacks affecting the Internet, the security solutions for detecting and preventing Internet-based DoS attacks are not always applicable to DoS attacks threatening WSNs. This is due to reasons such as resource and energy constraints, the open nature of wireless communications, and the dynamic topology of WSNs [1,8,9]. In addition, attackers continue to create new DoS attacks that exploit the structural and protocol vulnerabilities of evolving WSNs, including SDWSNs. Bin-Yahya et al. [10] described the most typical threats to SDWSNs as shown in Figure 4. In the 'flood' scenario, the malicious node (MLN) sends a large number of packets to its own cluster nodes as well as to the upstream nodes. This consumes the corresponding link bandwidth and node power, negatively affecting network performance. In the 'drop scenario', the MLN discards packets passing through it, preventing these packets from reaching the sink node. The data received by the DB server are incomplete, which may negatively affect the data processing output.

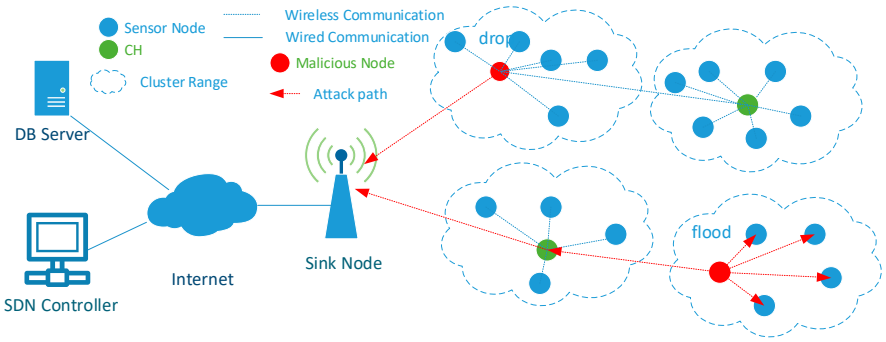

Figure 4. Examples of DoS attacks on an SDWSN.

Ganeriwal et al. [5] suggested that establishing a reputation-based system within a WSN (similar to reputation-based communication systems in human societies) can be used as an effective protection mechanism. The reputation-based framework proposed in their study provides security controls to counter the impact of compromised nodes (malicious or faulty).

A trust model establishes reputation-based trust relationships among the entities (nodes) of a network. This enables an entity to predict the future behavior of other entities based on knowledge about their reputation and supports decision-making in otherwise uncertain situations. Trust models are inspired by human society, where a person develops perceptions about the reputation of every other person they interact with. A reputation is built gradually through continuous observation and interaction [5].

Govindan et al. [11] defined trust in WSNs as the subjective assessment by one node of another node for reliability and accuracy in receiving or transmitting information under specific circumstances. In this study, we define trust as the subjective assessment of whether one node's behavior is normal (and therefore can be trusted) or not by another node, based on previous behavioral observations.

Trust evaluation in WSNs, also known as trust computation, involves calculating a truth value for each network node based on trust evidence. The trust value of a node provides a measure of its trustworthiness. A node evaluates the trustworthiness of other nodes using trust evidence gathered through one or more of the following methods: (i) direct evaluation, which is based on the observed behavior of the evaluated node; (ii) indirect

evaluation, which is based on recommendations for the evaluated node made by other network nodes, and (iii) iterative updates of historical and current evaluation values [11,12].

The numerical measurement of a piece of trust evidence is known as the 'trust metric'. A number of trust evidence collection and evaluation methods have been explored, including, for example, Pearson's correlation coefficient [13], link quality indicator (LQI) analysis [14], threshold-limiting methods [10,15,16], and Bayesian beta methods [14,17,18].

1.3. Research Problem and Questions

A number of studies have surveyed the application of trust models in WSNs. For instance, Muzammal et al. [19] explored IoT and RPL (routing protocol for low-power lossy networks) security. The authors analyzed the plausible attacks and suggested the use of trust models for routing protocol security. Alhandi et al. [20] conducted a review of trust assessment models for WSNs as part of an IoT security framework. Trust management techniques were examined by Tyagi et al. [21], who emphasized the potential of incorporating modern technologies like blockchain and machine learning (ML). Zhu et al. [22] focused on underwater WSN (UWSN) security and provided a classification of the security methods used. A recent review of trust models and approaches in IoT provides a managerial perspective on IoT security, including WSN security [23].

In empirical research, studies have focused on specific DoS attack scenarios. For example, Rahamathullah et al. [24] proposed a lightweight trust-based security system for preventing distributed DoS (DDoS) attacks affecting RPL. The energy-efficient trust management and routing system developed by Wang et al. [4] mitigates selective forwarding and new flow attacks in SDWSNs. Additionally, most trust models are only evaluated through simulations, and their application and feasibility in real-world scenarios are unknown. Therefore, there is a need to develop a better understanding of the challenges facing the development of feasible implementation applicable to the real-world trust models that can effectively address a broad spectrum of DoS attacks and protect the integrity of the services provided by WSNs.

This study seeks to establish a foundation for creating robust trust models that can identify and protect WSNs from a range of different types of DoS attacks by identifying the requirements for trust evidence gathering and the feasible trust evaluation approaches. This study examines the literature on trust models in WSNs to address the following research questions:

RQ1: What types of DoS attacks in WSNs can be addressed using trust models?
RQ2: What trust evidence is required to identify DoS attacks effectively?
RQ3: What approaches can be used to extract the required trust evidence?
RQ4: What methods for trust evaluation can be used?
RQ5: What are the key challenges to applying trust models for the defense of WSNs against DoS attacks?

1.4. Study Contributions

This study makes the following contributions:

1. This study develops a comprehensive classification of the types of DoS attacks in WSNs that can be addressed using trust models, including their key attack features and impact mechanisms.
2. By analyzing the key features of these DoS attacks, we identify the trust evidence that trust models are required to recognize DoS attacks effectively.
3. This study identifies, analyses, and compares the methods for extracting trust evidence.
4. This study provides a comprehensive summary of trust evaluation methods.

5. This study identifies the key challenges in applying trust models for the defense of WSNs against DoS attacks.

The remainder of this paper is organized as follows: The next section presents the methods used to identify the relevant research work used in this study. The Results Section classifies the types of DoS attacks in WSNs that can be addressed using trust models. It identifies the requirements for trust evidence, compares and analyzes various methods for extracting trust evidence, and summarizes trust evaluation methods. The Analysis and Discussion Section elaborates on the challenges in the deployment of trust models to counter DoS attacks in WSNs. The Conclusion Section summarizes this work, outlines its limitations, and suggests avenues for further research.

2. Methodology

We conducted a comprehensive literature review to identify, select, and appraise critically the extant research relevant to the research questions. We adapted the literature review methodology proposed in [25] as it offers a structured method for the identification and analysis of the extant research in a specific area.

2.1. Inclusion and Exclusion Criteria

As shown in Table 1, we chose studies published in English that concentrated on trust models targeting DoS attacks in WSNs. Topics in the field of SDWSNs were also included since SDWSNs are an evolving trend in developing optimized WSN architectures [26]. We excluded articles related to proposed artificial intelligence (AI) and blockchain solutions since such solutions are not yet feasible to be implemented in resource-constrained sensor nodes. As we were interested in the specific technical details and characteristics of the trust models proposed in the prior research, literature reviews were also excluded.

Table 1. Inclusion and exclusion criteria.

Criteria	Description
Inclusion criteria	Studies focusing on trust models addressing DoS attacks in WSNs. Studies focusing on trust models addressing DoS attacks in SDWSNs. Peer-reviewed articles, and conference papers. Studies published in English.
Exclusion criteria	Studies focusing on artificial intelligence or blockchain solutions for WSNs. Review papers. Non-peer-reviewed sources. Studies without available full text.

2.2. Search Strategy

The search repositories included SpringerLink, ACM Digital Library, IEEE Xplore, and ScienceDirect; we also explored Google Scholar. We used the following search terms: (“trust model*” OR “trust management”) AND (“wireless sensor network*” OR “WSN*” OR “sensor network*”) AND (“Denial of Service attack*” OR “DoS attack*”).

2.3. Screening and Selection Process

A flowchart of the screening process and selection process is shown in Figure 5. We retrieved a total of 639 relevant documents from the five repositories.

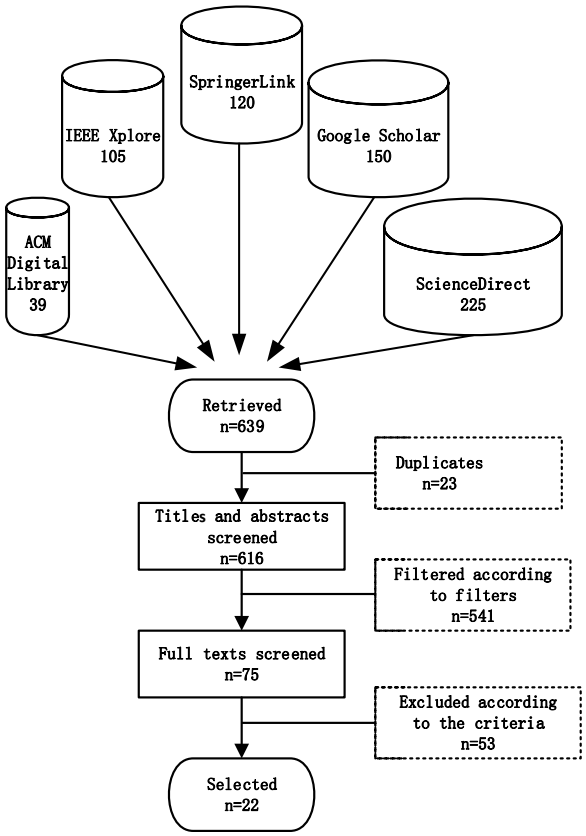

Figure 5. Screening and selection process.

Due to the advantages of RAYYAN [27], such as deleting duplication data automatically, keyword filtering, and the ability to use the mobile application offline, we chose RAYYAN as our screening tool. After excluding duplicates, the remaining 617 studies were screened based on the title and abstract. For title- and abstract-based filtering, we applied several filters in RAYYAN. Keywords for inclusion were "dos", "denial of service", and "trust", while exclusion keywords included "machine learning", "artificial intelligence", "blockchain", "survey", and "review".

After applying the filters, the number of relevant studies was reduced to 75. During the full-text-based filtering, we excluded articles based on the criteria in Table 1. The full-text screening resulted in a selection of 22 studies (15 journal articles and 7 conference papers), which were included in this review. These were published between 2004 and 2024, with 10 published in 2020–2024 (2 published in 2023–2024).

Since our search string only included "DoS attack" and excluded other types of attacks, studies investigating attacks such as on–off attacks or bad-mouthing attacks [28] were not selected. Additionally, we excluded studies involving the use of AI and blockchain technology. The screening results included a relatively small number of recently published papers (2023–2024). Therefore, we searched further for relevant research papers published from 2023 onward by tracing the citations and references in the studies already selected. Following this approach, we added eight new papers to the sample: these papers were published in 2023 ([3,28,29]), and five papers were published in 2024 ([2,30–33]).

2.4. Data Extraction and Synthesis Methods

From each study reviewed, we extracted information about the authors, the year published, the type of trust evidence used by the trust evaluation model, the use of direct or indirect trust evidence, the method for calculating trust value, the representation and the range of trust values, the structure of the trust model (distributed or centralized), the

model's capability to detect and/or defend against DoS attacks, the types of DoS attacks addressed, the simulation tools used, and the intended application area of the trust model. The information is collated in Appendix A, Table A1.

3. Key Technologies in Trust Management for WSNs

We conducted a detailed analysis of each study in the literature reviewed and applied a comparative analysis approach to investigate the specific research questions formulated in Section 1.2. As listed in Appendix A, Table A1, the majority of the reviewed models focus on traditional WSNs, as seen in studies such as [5,7,13–18,24,34–43]. While SDWSNs enhance the flexibility of the WSN and its management by separating network control from data forwarding and allowing the network to be dynamically adjusted and optimized according to actual needs [26,44], only four studies considered the use of trust models ([4,10,45,46]).

3.1. DoS Attacks in WSNs

First, we answer RQ1: What types of DoS attacks in WSNs can be addressed using trust models? The review of prior research identified several types of DoS attacks in WSNs (Table 2). These attacks mostly impact the data or control packet sending rate (PSR), data or control packet receiving rate (PRR), data or control packet forwarding rate (PFR), and energy consumption rate (ECR). For example, a distributed DoS (DDos) attack involves several compromised nodes launching a DoS attack simultaneously [24]. Low-rate DoS (LDoS) attacks against SDWSNs [43] occupy network resources by forcing nodes to process intermittently sent, low-rate traffic. The aim is to overload the network by sending messages that do not match the rules in the nodes' flow tables [4,10]. When a node receives such a message, it sends a new-flow control message to the network controller to request a new routing rule. The high volume of new-flow packets negatively affects the performance of the network.

Table 2. Types of DoS attacks in WSNs that can be addressed by using trust models, impact mechanisms, and key impact features.

Types of DoS Attacks	Impact Mechanisms	Key Impact Features
1. Selective Forwarding Attack [4]	MLNs affect the availability of the network services by dropping to a targeted destination node.	• Decreased PFR at the MLN. • Decreased ECR at the MLN.
2. Blackhole Attack [10]	The MLN claims to have the best route, causing all packets to be forwarded to it. It drops all forwarded packets.	• Increased PRR at the MLN. • Zero PFR at the MLN.
3. Grey Hole Attack [47]	A variant of the blackhole attack. The MLN drops packets intermittently or selectively. The degradation of network performance is less detectable.	• Increased PRR at the MLN. • Decreased PFR at the MLN.
4. Flooding Attack [18]	The MLN sends a large number of connection requests or packets and forces the attacked nodes to exhaust their resources by processing an excessive number of invalid requests.	• Increased PSR at the MLN. • Increased ECR the MLN.
5. Sinkhole Attack [47]	The MLN falsifies routing information and causes network congestion by attracting a large amount of network traffic.	• Increased PRR at the MLN.

Table 2. *Cont.*

Types of DoS Attacks	Impact Mechanisms	Key Impact Features
6. Sybil Attack [5]	The attacker disrupts the topology of the network by adding multiple fake nodes, which reduces the efficiency of the routing and data transmission.	• The appearance of multiple new nodes.
7. Vampire Attack [16]	The MLN manipulates network routing protocols to direct packets along longer or circular paths; legitimate nodes consume more energy and fail prematurely.	• Increased ECR at legitimate nodes.
8. DDoS Attack [24]	An enhanced version of the DoS attacks above; launched by multiple MLNs at the same time.	• As above, the attack originates from multiple MLNs.
9. LDoS Attack [43]	The quality of service of the network is eroded slowly by sending intermittent, low-rate malicious traffic. It is harder to detect.	• Slightly increased PSR at the MLN. • Slightly increased ECR at the MLN.
10. Hybrid DoS Attack [13]	Combines multiple attacks at the same time, such as sending malicious traffic while selectively dropping some legitimate packets.	• It depends on the type of attacks in the mix.
11. On–Off Attack [35]	The MLN switches randomly from attack to normal operation to prevent detection.	• Switching behavior at the MLN.
12. New-Flow Attack [4]	A flooding attack targeting the control plane of SDWSNs. The large number of new-flow control messages degrades network performance.	• Increased PSR at the MLN. • Increased ECR at the MLN.
13. Bad-Mouthing Attack [5]	The MLN spreads false negative information about legitimate nodes, causing their trust value to degrade.	• False trust information generated at the malicious node and propagated across the network.
14. Good-Mouthing Attack [5]	The MLN spreads false positive information about other MLNs.	• False trust information generated at the malicious node and propagated across the network.

In a Sybil attack, an MLN creates numerous fake identities, which disrupts the network communication channels [5]. Vampire attacks drain the energy of the sensor nodes, depleting their limited power supply [16]. Flooding attacks overwhelm nodes with excessive requests or data, exploiting network bandwidth and resources [18]. Hybrid attacks combine various attack strategies, increasing their complexity and effectiveness [13]. On–off attacks manipulate trust evaluation mechanisms by switching between legitimate and malicious behaviors [35]. Bad-mouthing and good-mouthing attacks use the trust model itself to disrupt WSN operations. The target nodes of a bad-mouthing attack become isolated and cannot participate in the network activities. In the good-mouthing attack, the attacker attempts to camouflage other malicious nodes by deliberately manipulating their reputation [5,10,15,35]. Finally, we added to the table grey hole attacks and sinkhole attacks in

WSNs, as discussed by Webber et al. [47], because their key impact features are similar to the ones of blackhole attacks.

3.2. Trust Evidence

In this section, we answer RQ2: What trust evidence is required to identify DoS attacks effectively? In resource-constrained WSNs, collecting large amounts of trust evidence leads to increased computational load, storage and communication overheads, and higher network latency [4]. To identify the trust evidence sufficient for DoS identification, we analyzed the trust evidence used by each of the trust models in Appendix A, Table A1, and the main impact features of DoS attacks on the network in Table 2.

As shown in Table 3, the trust evidence used in the majority of the trust models reviewed belongs to two categories: communication trust evidence and energy trust evidence. Communication trust evidence includes the PSR, the PRR, and the PFR. Energy trust evidence is provided by the ECR.

Table 3. Trust evidence for trust models in WSNs.

Trust Evidence Category	Trust Evidence
Communication trust evidence [10]	Sending rate—data packets Sending rate—control packets Receiving rate—data packets Receiving rate—control packets Forwarding rate—data packets Forwarding rate—control packets
Energy trust evidence [14]	Energy consumption rate
Data trust evidence [14]	Data accuracy

Notably, some trust models are used as evidence of additional trust features, such as the packet loss rate or the packet delivery success rate between two nodes [15,39,40]. For the purposes of this study, we consider these metrics as link quality indicators rather than as trust metrics (see Section 4 for a further discussion).

A comparison of Tables 2 and 3 indicates that the trust evidence in Table 3 would be sufficient to identify ten of the attack types in Table 2. The exceptions (the Sybil attack, the on–off attack, and the bad/good-mouthing attacks) are not manifested explicitly in higher energy consumption or noticeable changes in packet transmission rates. However, these attacks can be countered through adjustments to the trust value, as shown in Section 3.4.

As the main purpose of a WSN is to collect real-time data for further analysis [8], the accuracy of the data is critical: even if the network traffic characteristics and the energy consumption indicators at the WSN nodes are normal, the network is still unreliable or practically unavailable if the collected data are incorrect or not real-time. To include this quality requirement, we added 'data trust evidence' as a third category in Table 3; the data trust evidence is data accuracy (DA).

3.3. Approaches to Extracting Trust Evidence

In this section, we address RQ3: What approaches can be used to extract the required trust evidence? We examined the trust evidence extraction methods in the reviewed models and performed a comparative analysis of these methods, as shown in Table 3.

3.3.1. Extracting Packet Sending Rate

Most models use the observing node's promiscuous mode of 'listening' to extract this type of evidence. When the observing node detects data or a control packet originating from a particular node, the respective sent packet counter adds one to the count. In the

promiscuous receiving mode, any packet that arrives within its receiving range (sent either to itself or to another node) is either processed or discarded based on its destination address. The statistical information in the node's flow table is also updated. In addition to listening, Bin-Yahya et al. [10] employed a direct interaction approach in which the number of packets sent by a node is determined based on the direct interactions between the two nodes. This approach also involves retrieving interaction statistics from the flow table's statistical section.

3.3.2. Extracting Packet Receiving Rate

Among the models we reviewed, only two models used the receiving rate of data packets or control packets [15,35]. Both models used the packet receiving rate to calculate the packet loss rate. The extraction of the packet receiving rate was performed by tracking the packets received by the node during direct interactions.

We propose that the packet receiving rate can be used to detect attacks that attract traffic, such as blackhole and sinkhole attacks. The approach can be summarized as follows. The sensor node works in promiscuous receiving mode. By analyzing the destination address and type of each received packet, the node to which the data or control packet was sent can be identified, and the corresponding count can be increased accordingly.

3.3.3. Extracting Packet Forwarding Rate

Most models used listening and watchdog mechanisms to extract the packet forwarding rate. Bin-Yahya et al. [10] employed two variations of the listening and watchdog methods: one where the sender listens to the receiver, and another where a third party simultaneously listens to both the sender and receiver to gather evidence of the forwarding rate. They also used a method to determine the forwarding rate of control packets via ACK messages in SDWSNs. We believe the third-party method, which monitors both the sender and receiver simultaneously, is redundant with the sender-listening method and adds unnecessary complexity. For instance, a sensor node would need to initiate a watchdog for both the data packets it sends and the source data packets it overhears. Additionally, the issue of duplicate records must be addressed.

Anwar et al. [35] mentioned extracting the forwarding rate by querying the statistical information from the traffic profiles of neighboring nodes. However, this method depends on interactions involving request and response messages related to traffic profiles, resulting in increased communication overhead.

Based on these observations, we summarize the approach for extracting the forwarding rate of data or control packets as follows. Sensor nodes enable the promiscuous receiving mode and start a watchdog timer after sending a packet. If the destination node successfully forwards the packet before the watchdog expires, the successful forwarding count of the destination node is incremented by one; otherwise, the failed forwarding count is incremented by one.

The ACK-based method for extracting the control packet forwarding rate can serve as a supplement to the listening method. If either the forwarded packet or the ACK message is received by a sender during the monitoring period, it can be considered as a successful forwarding.

3.3.4. Extracting the Energy Consumption Rate

We identified two methods for extracting the energy consumption rate. Wu et al. [14] calculated a node's energy consumption rate based on the residual energy parameter in the beacon message. Jinhui et al. [13] converted the energy consumption sequence of a sensor node during an operation cycle into a power consumption sequence. However, the second method relies on sensor nodes monitoring their own energy consumption with

high precision. Given the large number and low cost of sensor nodes, achieving such high precision is not very realistic.

3.3.5. Extracting Data Accuracy

We examined how the data accuracy was extracted in the reviewed models. There are two main-stream approaches: the first approach involves extracting the data sequences collected by the evaluating node and the evaluated node separately and using a probability distribution method to assess the differences in the data sequences [14]. The second approach involves collecting all the data from neighboring nodes through the sensor node's promiscuous receiving mode and using an outlier detection algorithm to verify the accuracy of the data [5].

The points made above are summarized in Table 4. The table lists the methods identified in the literature and shows the approach considered as the most efficient.

Table 4. Methods for extracting trust evidence.

Trust Evidence	Extraction Methods	Comparison
PSR	Direct interaction.	Included in method 'nodes' promiscuous receiving mode.
	Nodes' promiscuous receiving mode.	
	Retrieve from flow table.	The statistical information in the flow table is derived using the method 'nodes' promiscuous receiving mode.
PRR	Direct interaction.	Included in method 'nodes' promiscuous receiving mode.
	Nodes' promiscuous receiving mode.	
PFR	Watchdog plus sender overhearing receiver.	
	Watchdog plus third-party overhearing sender and receiver.	Redundant (with method 'watchdog plus sender overhearing receiver' selected); increases complexity.
	Using ACK messages.	Watchdog plus method 'watchdog plus sender overhearing receiver'.
	Using traffic profiles.	Relies on interactions involving request and response messages related to traffic profiles, leading to communication overhead.
ECR	Calculate the energy consumption rate based on the beacon message. Convert the energy consumption sequence into the power consumption sequence.	Based on sensor nodes monitoring their own energy consumption with high precision, not very realistic.
DA	Extract differences in data sequences of the evaluating node and the evaluated node.	Less accurate than the method 'extract the accuracy of the data using an outlier detection algorithm based on all neighborhood data references'.
	Extract the accuracy of the data using an outlier detection algorithm based on all neighborhood data references.	

3.4. Trust Evaluation Methods

In this section, we seek the answer to RQ4: What are the methods for trust evaluation? As stated earlier, trust evaluation includes direct trust evaluations, indirect trust evaluations, and updating the trust value. Based on the characteristics of trust evidence, direct trust evaluation methods can be divided into two categories: threshold-limiting methods and success–failure methods [10]. The threshold-limiting method can be used for evaluating

packet sending and receiving rates and the energy consumption rate, while the success–failure method can be used for evaluating the packer forwarding rate and the data accuracy.

3.4.1. Direct Trust Evaluation

Threshold-limiting methods

In the models we reviewed, all evaluations of the packet-sending rate and some evaluations of the energy consumption rate used the threshold-limiting method. However, the methods for setting thresholds and calculations varied. One of the simplest methods, and the one used by most models, is to set an upper limit threshold value. Once the evaluated metric exceeds this upper limit, the node is considered untrustworthy. For example; in the trust-based technique for monitoring medical implants [7], the calculated trust value is binary, either 0 (untrustworthy) or 1 (trustworthy). The calculation method is shown in Equation (1)

$$T = \begin{cases} 0, & D_{rate} \geq th \\ 1, & D_{rate} < th \end{cases} \tag{1}$$

where T represents the trust value, D_{rate} represents the data packet sending rate, and th represents the upper limit of the data packet sending rate. The threshold is set based on the patient's environment (home, workplace, public places) and the specific metrics of different implants (heart rate, body temperature, blood pressure, etc.).

In ETMRM [4], the evaluation of the new flow-sending rate still uses an upper limit threshold. However, the trust value is not binary; instead, it decreases as the packet-sending rate increases. The calculation method is shown in Equation (2):

$$PT_{ij}(t) = \frac{1}{\left\lceil \frac{invl_j}{F_\tau} \right\rceil} \tag{2}$$

where F_τ is the maximum number of new flows a node can receive from one of its neighbors during a period t. It adapts to the actual network environment. $invl_j$ is the number of new flows node j sends during period t.

In the TSW model [10], the upper limit threshold is divided into three levels, causing the trust value to decrease at different rates within different threshold intervals. The calculation method is shown in Equation (3):

$$T_{x,y}^{Metric} = \begin{cases} 1, & N_y \leq \eta 1 \\ 1 - \dfrac{\eta 1 - N_y}{2(\eta 1 - \eta)}, & \eta 1 < N_y \leq \eta \\ \dfrac{N_y - \eta 0}{2(\eta - \eta 0)}, & \eta < N_y \leq \eta 0 \\ 0, & othwise. \end{cases} \tag{3}$$

where N_y represents the number of new flows send by node y and η represents the maximum number of expected data and packet-in messages from the sending node. The threshold points $\eta 1$ and $\eta 0$ are calculated as follows: $\eta 1 = \eta/2$ and $\eta 0 = \eta + \eta/4$.

We set the upper limit threshold value of the above three calculation methods to 50 and plotted the function curves of the three algorithms for comparison, as shown in Figure 6.

From the above figure, we can see that the methods corresponding to Equations (1) and (2) maintain a state of trust until the evaluated metric reaches the threshold. After exceeding the threshold, the method corresponding to Equation (1) sets the trust value directly to 0, while the method corresponding to Equation (2) decreases the trust value gradually in a stepwise manner. In contrast, the method corresponding to Equation (3) starts to

decrease the trust value linearly as the evaluated metric approaches the threshold, and after exceeding the threshold, it decreases linearly at a faster rate until it reaches 0. Based on the comparison, the third method responds more promptly and sensitively to changes in trust metrics.

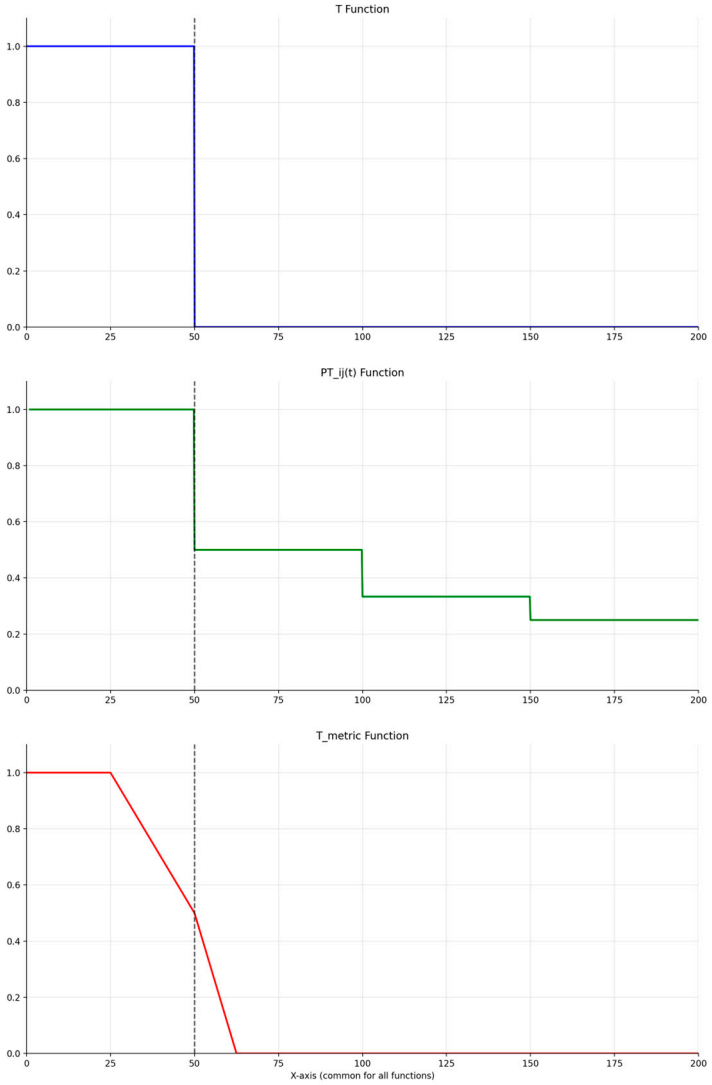

Figure 6. Comparison of three threshold-limiting methods.

Success–failure methods

The two most common approaches in success–failure methods are Bayesian beta methods and entropy-based methods. Ganeriwal et al. [5] analyzed, in detail, the applicability of Bayesian formulations and beta distributions to WSN trust systems. The calculation of trust values using the Bayesian beta method is shown in Equation (4):

$$T_{ij} = E(R_{ij}) = E(Beta\{\alpha_j + 1, \beta_j + 1\}) = \frac{\alpha_j + 1}{\alpha_j + \beta_j + 2} \quad (4)$$

where α_j and β_j represent the cooperative and noncooperative interactions between nodes i and j, respectively (from the perspective of node i).

Bin-Yahya et al. [10] enhanced the Bayesian formulation by incorporating reward and penalty factors. These factors reward or penalize based on the number of successes or failures, as shown in Equation (5):

$$T_{x,y}^{Metric} = \frac{N_{x,y}^{+} + 1}{N_{x,y}^{total} + 2} \cdot \frac{N_{x,y}^{+}}{N_{x,y}^{+} + 1} \cdot \frac{1}{\sqrt{N_{x,y}^{-} + 1}} \tag{5}$$

where x is the evaluating node, which calculates the trust value and counts the number of success and failure hits, and y is the evaluated node by node x. $N_{x,y}^{+}$ is the successful count that node x recorded about node y. $N_{x,y}^{-}$ is the number of failure hits that node x experiences with node y. $T_{x,y}^{Metric}$ is the trust metric computed by x for y. In Equation (2), the first term is the Bayesian factor, followed by the reward and penalty terms. The rewarding factor ensures a gradual increase in the trust value, whereas the penalty factor ensures a rapid decrease in the trust value after failure.

Khan and Singh [3] used penalty and reward parameters that can be flexibly adjusted according to the network and the application, as shown in Equation (6):

$$T_{x,y}(\Delta t) = \left[4 \times \left(\frac{S_{x,y}(\Delta t)}{S_{x,y}(\Delta t) + U_{x,y}(\Delta t)} \right) * \frac{1}{\sqrt{\mu * U_{x,y}(\Delta t) + 1}} * \varnothing^{S_{x,y}(\Delta t)} \right] \tag{6}$$

where $T_{x,y}(\Delta t)$ represents the trust value evaluated by node x for node y within the time window Δt. $S_{x,y}(\Delta t)$ and $U_{x,y}(\Delta t)$ represent the number of successful and unsuccessful interactions between node x and node y, respectively. μ and $\varnothing^{S_{x,y}(\Delta t)}$ are the adjustable penalty and reward parameters, respectively. The value of $\varnothing^{S_{x,y}(\Delta t)}$ increases with the number of successful interactions and can be adjusted to reflect the network and the application. Han et al. [32] similarly added a penalty factor in the model they proposed.

The entropy trust model uses an entropy function to calculate the trust value based on the Bayesian beta method so that the trust value is not a linear function of probability, as shown in Equations (7) and (8) and Figure 7:

$$T_{Entr} = \begin{cases} 1 - H(p), & for\ 0.5 \leq p \leq 1; \\ H(p) - 1, & for\ 0 \leq p < 0.5. \end{cases} \tag{7}$$

$$H(p) = -plog_2 p - (1-p)log_2(1-p) \tag{8}$$

where p is the trust value calculated using the Bayesian beta method and $H(p)$ is the entropy function.

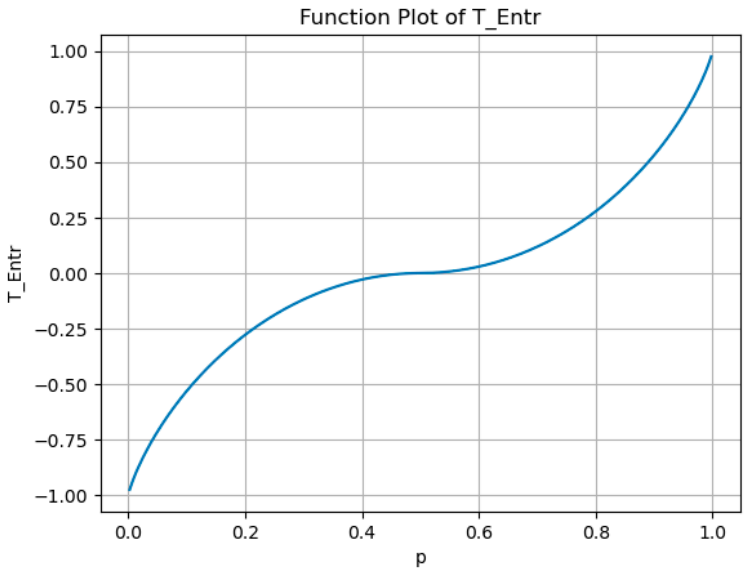

Figure 7. Function plot of T_{Entr}.

Cho and Qu [37] applied source-level trust assessment based on the Bayesian beta method and the entropy-based method. Through mathematical analysis and simulation experiments, they verified that in selective forwarding attack scenarios, it is much more difficult to obtain a high trust value in trust models using the entropy-based method trust model compared to trust models using the Bayesian beta method. This is because, with the entropy-based method, nodes need to have a very low packet loss rate to obtain a high trust value. The entropy-based method trust model allows faster identification of attack victims as the increased number of lost packets causes the trust value to decrease further. However, the computational complexity of the entropy-based method is higher than that of the Bayesian beta method.

A form of direct trust evaluation can be used to counter a Sybil attack. In Sybil, the attacker forges multiple new identities that appear as new nodes. As WSNs are open by nature and normally have a high node failure rate, the addition of new nodes is a common occurrence. Therefore, it is not easy to determine whether new nodes are fake or legitimate entities. To mitigate the risk, Ganeriwal et al. [5] proposed to initialize all new nodes (physical or virtual) as 'untrustworthy'. The trust value of a legitimate new node will increase as the node successfully participates in transactions.

3.4.2. Indirect Trust Evaluation

Indirect trust evaluation helps improve the accuracy of the trust evaluation and accelerates the convergence time of the trust value calculations. Nodes can quickly obtain trust information about a specific node by leveraging the experiences of other nodes in the network. This allows us to build the trust system of the entire network in a timely manner. Table 5 shows a comparison of the indirect trust evaluation methods used in the trust models reviewed.

Dempster–Shafer

Each node possesses both direct and indirect trust values for its neighbor nodes. When a node shares its trust value of a specific neighbor with other nodes, the question arises whether it should share the final combined trust value, which includes both direct and indirect trust, or only the direct trust value. In the studies we reviewed, only Ganeriwal et al. [5,38] analyzed this issue and noted that to avoid the same trust information circulating back to its origin point, nodes should broadcast only their direct trust values to their neighbors. They applied the Dempster–Shafer belief theory to resolve the integration of direct and indirect trust metrics. The closed-form expression for the new trust metric after trust integration is shown in Equations (8) and (10):

$$\alpha_j^{new} = \alpha_j + \frac{2\alpha_k\alpha_j^k}{\left[(\beta_k+2)\left(\alpha_j^k+\beta_j^k+2\right)\right]+2\alpha_k} \tag{9}$$

$$\beta_j^{new} = \beta_j + \frac{2\alpha_k\beta_j^k}{\left[(\beta_k+2)\left(\alpha_j^k+\beta_j^k+2\right)\right]+2\alpha_k} \tag{10}$$

where $(\alpha_j,\ \beta_j)$ represents the trust metric of node j maintained by node i before integration, $(\alpha_k,\ \beta_k)$ represents the trust metric of node k maintained by node i, $\left(\alpha_j^k,\ \beta_j^k\right)$ represents the trust metric of node j received from node k by node i, and $\left(\alpha_j^{new},\ \beta_j^{new}\right)$ represents the trust metric of node j maintained by node i after integration.

Table 5. Classification and comparison of indirect trust evaluation methods.

Trust Models	Methods	Comparison
[5,38]	Dempster–Shafer belief theory method	Suitable for trust models that propagate trust metrics; has strong theoretical support
[15,16,24,31,35,42]	Arithmetic mean method	Suitable for trust models that propagate trust values and do not consider the trustworthiness of the recommendation information; accuracy is low
[14,17,18,32]	Weighted average method	Suitable for trust models that propagate trust values and consider the trustworthiness of the recommendation information based on the recommender's trust value; accuracy is moderate
[10,30,34,45]	Outlier detection	Suitable for trust models that propagate trust values, using outlier detection methods to evaluate the trustworthiness of the recommendation information; accuracy is high
[39]	Forgetting curve method	Suitable for trust models that propagate trust values and do not consider the trustworthiness of the recommendation information; accuracy is low
[46]	Intrusion detection system	Uses the IDS to detect network anomalies from the network statistical data
[33]	Method based on link quality and node reliability	Suitable for trust models that propagate trust values and do not consider the trustworthiness of the recommendation information; accuracy is low
[2]	Selective re-evaluation method	Risk assessment values are not involved in the calculation of trust values, which avoids bad-mouthing attacks, but re-evaluation increases latency and burden

Arithmetic Mean

In the models we reviewed, except for RFSN, all the models that include indirect trust directly propagate the calculated trust values during the indirect trust propagation process, rather than propagating trust metrics. These models generally use the arithmetic mean method to integrate the trust values provided by all recommenders. Then, the direct trust value is weighted and averaged with the integrated indirect trust value [15,16,24,31,35,42], as shown in Equations (11) and (12):

$$IT = \frac{1}{k}\sum_{m=1}^{k} DT(m) \tag{11}$$

$$FT = \alpha \cdot DT + \beta \cdot IT \tag{12}$$

where IT represents the indirect trust value of evaluated node j, k represents the number of recommenders, $DT(m)$ represents the direct trust of node j evaluated by recommender m, and α and β are the weights for direct trust and recommended trust, respectively, with $\alpha + \beta = 1$. The choice of weights depends on the specific application.

Weighted Average

Other models use a weighted average method to integrate the trust values provided by all recommenders. Then, the direct trust value is weighted and averaged with the integrated indirect trust value [14,17,18], as shown in Equations (13)–(15):

$$rec\_Trust = \sum_{x=1}^{m} \varphi_x dir\_Trust_{rx}^{j} \tag{13}$$

$$\varphi_x = \frac{dir\_Trust_i^{rx}}{\sum_{x=1}^{m} dir\_Trust_i^{rx}} \tag{14}$$

$$int\_Trust = \alpha \cdot dir\_Trust + \beta \cdot rec\_Trust \tag{15}$$

where m denotes the number of recommenders, $dir\_Trust_{rx}^{j}$ represents the direct trust of node j evaluated by recommender rx, and φ_x denotes the weight of the direct trust recommended by recommender rx. $dir\_Trust_i^{rx}$ represents the direct trust of recommender rx evaluated by node i and α and β are the weights for the direct trust and recommended trust, respectively, with $\alpha + \beta = 1$. Additionally, Han et al. [32] used an improved density peak clustering (DPC) algorithm to identify and filter out nodes providing false recommendations before performing indirect trust evaluation using the weighted average method.

Outlier detection

Some models [10,34,45] use similar methods as those mentioned above but use outlier detection methods to calculate φ_x. They assign weight based on the degree of difference between the trust value given by a particular recommender and the average trust value given by all recommenders, as shown in Equation (16):

$$\varphi_x = 1 - \left| dir\_Trust_{rx}^{j} - \frac{\sum_{x=1}^{m} dir\_Trust_{rx}^{j}}{m} \right| \tag{16}$$

Ye and Jiang [30] used a Gaussian distribution outlier detection algorithm to calculate the reliability of each recommendation value. Recommendation values that deviate from the mean by more than twice the standard deviation are considered outliers and are excluded. The reliability of the recommendation values after excluding outliers is shown in Equations (17) and (18)

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)} \tag{17}$$

$$T = 1 - 2\left| \int_{\mu}^{X} f(x)dx \right| \tag{18}$$

where $f(x)$ is the probability density function of a Gaussian distribution. Assuming the set of recommended trust values follows a Gaussian distribution, μ is the mean of this set and σ is the standard deviation of this set. T is the reliability value of the recommended trust value X.

Forgetting Curve

Gautam and Kumar [39] used a forgetting curve to balance the values of direct trust and indirect trust. The calculation methods are shown in Equations (19)–(21):

$$T_{idt}(i,j) = \sum_{k=1}^{n} \frac{P_{k,j}}{P_{k,j} + N_{k,j}} \tag{19}$$

$$T_{i,j} = c \cdot T_{dir}(i,j) + (1-c) T_{idt}(i,j) \tag{20}$$

$$c = e^{-\beta \frac{\Delta t}{\gamma}} \tag{21}$$

where $T_{dir}(i,j)$ represents the direct trust value, $T_{idt}(i,j)$ represent the indirect trust value, $P_{k,j}$ represents the positive recommendation from node k to node j, $N_{k,j}$ represents the negative recommendation from node k to node j, c is the forgetting curve function, Δt is the time difference between the evaluation period, β is time decay factor, and γ is the cycle in the forgetting curve. As shown in Figure 8, different parameter combinations (β and γ) lead to varying changes in the value of c over time.

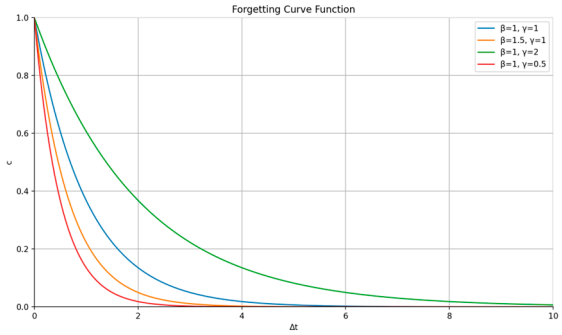

Figure 8. Function plot of the forgetting curve.

Intrusion Detection System

At the controller level, Isong et al. [46] used an intrusion detection system (IDS) module to analyze network statistics collected from each node to complete indirect trust evaluation. When the IDS detects network anomalies from the statistical data and identifies a node associated with these anomalies, the identified node is considered vulnerable or compromised. Consequently, the untrustworthiness of such a node increases by one, and its packet forwarding activities are temporarily halted.

Method Based on Link Quality and Node Reliability

Almutairi et al. [33] calculated the weight of each recommender based on link quality and node reliability, as shown in Equation (22):

$$w_r = \frac{1}{(1 + (\alpha R_r + \beta L Q_r))} \tag{22}$$

where R_r is the node reliability of recommender r and LQ_r is the link quality between recommender r and the evaluator. Link quality is evaluated based on the signal-to-noise ratio, distance, and received signal strength. Node reliability is evaluated based on the success rate of interactions between nodes.

Selective Re-evaluation Method

In indirect trust evaluation, Saidi [2] used the following strategy to avoid false recommendations from malicious nodes. The recommending node sends the risk assessment

value of the evaluated node to the evaluating node. Based on the received assessment value, the evaluating node decides whether to implement stricter evaluations on the evaluated node. The risk assessment value is not included in the trust value calculation. It is calculated based on all malicious behavioral activities of the evaluated node in the network, including packet loss, data deviation, and energy consumption.

Indirect trust evaluation methods can be used to defend the network against attacks such as bad/good mouthing. In this type of attack, the attacking node generates and propagates false recommendations across the WSN. While it may be hard to determine whether a recommendation is truthful, recommendations from nodes with 'good' trust values should be more reliable. This way, the trust value of the recommending node can be used to assign a weighting to the recommendation [5]. To address the risk of a previously trustworthy node becoming compromised and starting to spread false trust information, receiving nodes compare the recommendations they receive. In the case of a significant deviation, a recommendation may be declared untrue [10]; the recommender's trust value will be decreased accordingly.

Indirect trust evaluation requires the sharing of trust information between nodes, which increases communication costs. As it should not affect normal data communication and should ensure that trust values are updated in a timely manner, it is suitable for the early stages of establishing the trust system. During stable operation, propagating trust information when there are significant changes in the trust values is a feasible approach.

3.4.3. Updating the Trust Value

We identified the models that use trust value update methods to address trust value 'aging' (Table 6).

Table 6. Classification and comparison of trust value update methods.

Models	Methods	Comparison
[2,3,5,14,17,18,28,29,38,40,41,45]	Weighted average	Do not consider poor past performance
[10]	Improved weighted average	Consider poor past performance
[30,33,39]	Time lapse factor	Do not consider poor past performance

Weighted Average

Most of the reviewed models [2,5,14,17,18,28,38,40,41,45] update the trust value using the weighted average method, as shown in Equation (23):

$$T(t) = \theta \cdot T(t-1) + (1-\theta) \cdot T(t) \quad (23)$$

where θ is the aging factor, $T(t-1)$ is the trust value of a particular node in the $(t-1)^{th}$ period, and $T(t)$ is the trust value of a particular node in the t^{th} period. Khan and Singh [3] considered the number of successful and unsuccessful interactions in the current evaluation period to update trust values. When both the number of successful and the number of unsuccessful interactions are greater than 0, the above weighted average algorithm is used. When the number of unsuccessful interactions is 0 and the number of successful interactions exceeds the threshold, the trust value is updated to the maximum value. When the number of successful interactions is 0, the trust value is updated to the minimum value.

Iftikhar et al. [29] considered the trust values of the previous two consecutive periods of trust value updates, as shown in Equation (24).

$$TT(n) = (\mu_{n-1} \cdot TT(\mathrm{n}-1) + \mu_n \cdot \mathrm{CT}(t)) + (\mu_{n-2} TT(n-2) + \mu_{n-1} CT(n-1))/2 \quad (24)$$

where $TT(n)$ represents the total trust value of the n^{th} evaluation period, $\mathrm{CT}(t)$ represents the comprehensive trust value of the n^{th} evaluation period, and θ is the aging factor: $\mu_{n-2} = \theta$, $\mu_{n-1} = 1 - \theta$, $\mu_n = 2 - \theta$.

Improved Weighted Average

Bin-Yahya et al. [10] used an improved trust update mechanism to give more weight to the historical trust value when the current trust value is greater than the historical trust value, as shown in Equation (25):

$$T(t) = \begin{cases} (1-\alpha)T(t) + \alpha T(t-\Delta t), & if\ T(t) \leq T(t-\Delta t) \\ (1-(\alpha+\beta))T(t) + (\alpha+\beta)T(t-\Delta t), & if\ T(t) > T(t-\Delta t) \end{cases} \quad (25)$$

where $T(t)$ is the new trust value computed for the current window Δt at time t, while $T(t-\Delta t)$ is the previous trust value calculated in the previous window at time $(t-\Delta t)$. α is an aging factor, and β is the newly defined aging factor, with $\alpha + \beta < 1$.

Time Lapse Factor

Gautam et al. [39] used a time lapse function to dynamically balance historical trust values and current trust values. The calculation methods are shown in Equations (26) and (27):

$$T^n = k \cdot ST + (1-k) \cdot T^{n-1} \quad (26)$$

$$k = \begin{cases} 1 - \left(\frac{t_{n-1} - t_1}{t_n - t_1}\right)^2, & If\ t_n > t_1 \\ 1, & otherwise \end{cases} \quad (27)$$

where ST represents the current trust value, T^{n-1} represents the previous trust value calculated in the previous window at time t_{n-1}, and k is the time lapse function. Ye et al. [30] and Almutairi et al. [33] used a similar method to update trust values, but they replaced the time decay function with a time decay factor specific to the context of their study (underwater scenarios).

Among the three methods, only the second method considers the impact of lower historical trust values. This means that nodes with poor past performance should be penalized by limiting the growth rate of their trust values. Therefore, the second method is the most effective one. For example, it can be used to defend against the on–off attack, where the attacking node or nodes evade detection by stopping the attack for a period and then resuming it. The punishment for past bad behavior will lead to a very rapid decrease in the trust value, making the malicious node untrustworthy and thus limiting the impact of the attack. As good past behavior is rewarded at a much slower rate, the malicious node will need more time before its trust value is sufficiently increased to be considered trustworthy.

4. Challenges

In this section, the analysis of the findings presented in the previous section is used to address RQ5: What are the key challenges to applying trust models for the defense of WSNs against DoS attacks?

4.1. Threshold Limits

Threshold-limiting methods require threshold limits to be set by the user in advance, usually based on expectations about network traffic, such as the maximum number of data and control packets transmitted during a given time period [10]. As WSNs are application-driven, their performance requirements vary with the application context. Without significant practical experience, identifying the threshold appropriate for each context is a challenge to the development of feasible trust models for WSNs.

Closely integrating the trust model into the fabric of the WSN may help address the issue. For example, rather than comparing a node's trust metric to the corresponding threshold, the trust model can include a comparison of the trust metrics of a suitable reference set of nodes and search for outliers. The detection of a flood attack provides an illustration: the evaluating node deploys a promiscuous mode to monitor all nodes within its neighborhood and record their packet sending rate. The resulting datasets are forwarded to an outlier detection algorithm, which identifies them as malicious nodes with abnormal packet sending behavior.

4.2. Weighting Trust Evidence Metrics

Trust assessment involves allocating specific weights to the various evaluation metrics used. Most trust models rely on the WSN user to determine the correct weights [10,18,30,35]. However, this approach is not practical as it requires an advanced understanding of the specific method used for trust evaluation. Thus, some authors use only the arithmetic mean method to integrate various trust metrics [2].

To meet the challenge, Wu et al. [14] proposed the following method for assigning weights: First, a trust threshold is predefined. If the trust values of the three trust metrics, i.e., the communication trust metric (CMT), energy trust metric (ET), and data trust metric (DAT), are either above or below this threshold, the weights are evenly distributed among the three metrics. If some of the trust metric values are above the threshold and some are below, the metrics below the threshold are evenly redistributed with weights, while the metrics above the threshold are ignored to prevent the masking of malicious activity. This approach prevents attacks manifested in one trust evidence aspect from being masked by good performance in other aspects. However, the proposed trust model includes only three trust evidence metrics and, therefore, addresses only some DoS attacks. Second, assigning equal weights to trust metrics is not very accurate as the functionalities and the performance expectations of WSNs may differ significantly (e.g., some networks require higher data accuracy, whereas others require high volumes of network traffic).

Almutairi et al. [33] used the softmax function based on ML principles to achieve dynamic weight allocation among various trust metrics. The weights of the trust metrics can be flexibly adjusted according to the differences between the trust metric values and the set thresholds, but the setting of thresholds and fine-tuning parameters still requires human intervention. Additionally, parameter selection is challenging.

Zhang et al. [28] used a method combining fuzzy theory and the analytic hierarchy process to achieve fuzzy comprehensive dynamic weight evaluation. This calculation process includes four steps. First, the three trust metrics CMT, DAT, and ET are mapped to six fuzzy subsets $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$, representing six different trust levels from complete distrust to complete trust. Second, the trapezoidal membership function is used to calculate the membership degree of each trust factor in the six fuzzy subsets, forming the membership matrix $E_{ij} = [e_{ij}]$. Third, a non-increasing function $\lambda_i(T_i)$ regarding trust values is introduced to achieve variable weights $w_i = \frac{\lambda_i}{\sum_{j=1}^{3} \lambda_j}$, i.e., when T_i is smaller, the weight w_i increases, ensuring that the impact of low trust factors is more significant. Additionally, it is necessary to set the basic weights w_{mi} (referring to the initial weights of each

trust factor under normal circumstances) and the upper weight limits w_{0i} (used to amplify the impact when a certain trust factor is too low). Finally, the comprehensive trust value is calculated based on the weight vector $W = \{w_1, w_2, w_3\}$ and the membership matrix E_{ij} as $V = W * E_{ij}$; the comprehensive trust value is de-fuzzified using the centroid method. This method involves analytic hierarchy fuzzy logic operations and matrix calculations. Its computational complexity is higher compared to traditional weighted average methods. Parameter tuning is also a challenge. We conducted a comparative analysis of the three methods for weight allocation to trust metrics and summarized their pros and cons (Table 7). As shown, none of the methods offers a lightweight reliable approach for dynamic weight allocation that does not require human intervention. This is still a significant impediment to the development of trust models for the protection of WSNs.

Table 7. Classification and comparison of three methods for weight allocation to trust metrics.

Author, Year	Weight Assignment Methods			Pros and Cons
Wu et al., 2019 [14]; Ahmed et al., 2024 [17]	Cases	Trust Values	Weights	Pros: prevents attacks manifested in one trust evidence aspect from being masked by good performance in other aspects Cons: only supports three trust evidence metrics; not very accurate
	1	$DAT \geq 0.5$ $CMT \geq 0.5$ $ET \geq 0.5$	$w_{dat} = 1/3$ $w_{cmt} = 1/3$ $w_{et} = 1/3$	
	2	$DAT < 0.5$ $CMT < 0.5$ $ET < 0.5$	$w_{dat} = 1/3$ $w_{cmt} = 1/3$ $w_{et} = 1/3$	
	3	DAT, CMT, or ET have mixed values greater and lower than 0.5	The weight of the trust factor that is greater than 0.5 will be assigned 0 and the remaining weights will be assigned equally	
Almutairi et al. [33]	$w_i = \frac{e^{\frac{(\theta_i - Trust_i)}{\alpha}}}{\sum_{m=1}^{M} e^{\frac{(\theta_i - Trust_i)}{\alpha}}}$			Pros: dynamic weight allocation Cons: requires human intervention; parameter selection is challenging
Zhang et al. [28]	Fuzzy subsets $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$; Membership matrix E_{ij}; Variable weights $w_i = \frac{\lambda_i}{\sum_{j=1}^{3} \lambda_j}$, the basic weights w_{mi}, and the upper weight limits w_{0i}; Comprehensive trust $V = W * E_{ij}$; De-fuzzify the comprehensive trust value using the centroid method.			Pros: dynamic weight allocation; prevents attacks manifested in one trust evidence aspect from being masked by good performance in other aspects Cons: requires human intervention, parameter selection is challenging, high computational complexity

4.3. Loss of Trust Information

Managing information loss refers to the problem of storing and distributing trust information effectively and efficiently. Trust models in WSNs can be distributed [5] or centralized [7]. In a distributed trust model, each node computes and stores its own trust value. If a node fails, the impact of the subsequent loss of trust information will be limited to the neighbor not being able to evaluate the failed node. A typical method for managing trust information in a distributed trust model is storing a local copy of the network at each node along with a node backlist [24]. When a node broadcasts a DIO message to its neighbors, the node appends to it its own trust value and the blacklist. A somewhat similar approach was proposed by Anwar et al. [35], which is also suitable for a distributed

trust model: WSN nodes record trust evidence in their traffic profiles and share them with each other.

In a centralized trust model, trust value computation occurs at the nodes acting as heads of node clusters or at the base station. Trust information is stored both at the cluster head (base station) and at the nodes. The trust evidence collected by nodes and/or preliminary trust evaluation results are sent to the respective cluster heads (base station). The cluster head (base station) stores these records, uses them to calculate, aggregate, and maintain the trust values of each member of the cluster, detects malicious nodes based on trust values, and takes countermeasures.

Several methods for recording and propagating trust information in centralized trust systems were proposed in the reviewed literature. In [16], the cluster head is responsible for gathering trust evidence from cluster members and forwarding it to the base station, which maintains a comprehensive trust table. A similar table is proposed in [13], which is created and updated at the cluster head. For SDWSNs, Wang et al. [4] proposed using the number of successfully forwarded/failed to forward data and control packets and the number of new flow packets as metrics to calculate the node's trust value locally and to forward it to an aggregator node.

While the reviewed studies discuss the advantages and the applicability of the proposed methods for storing and managing trust information, they do not discuss specifics, such as for how long the trust information should be stored. Are there any options for retrieving centrally stored trust information that has been lost? The second question is particularly challenging: in the case of irrevocable trust information loss in a centralized trust model, the trust system of the entire WSN will need to be re-built.

4.4. Link Quality

WSNs often operate in unattended and open environments, where the quality of the radio transmission can be affected by factors such as weather conditions. If the quality of the link between the evaluating node and the evaluated node is poor or unstable, packet loss may occur. This can result in calculating a low trust value for a non-malicious node. Assessing link quality ahead of trust value calculations may address the issue as the trust evaluation can be put on hold if the link quality is compromised and reattempted later when the link becomes stable and reliable again [14,18,28].

In wireless networks, the link quality indicator (LQI) is used as a measure of communication link quality. In the case of the most widely used radiofrequency, CC2420, the LQI value is embedded in the received data packet; it varies from 0 to 255. An active node assesses the quality of a link between itself and a target node, gathering the LQI for a period of time and calculating the average. The average is compared to the threshold, e.g., 220, to determine if the link is stable or not [14].

However, using the LQI in trust evaluation models may not be reliable due to the assumption of uniform network conditions. In environments where natural conditions may weaken the radio signal, this may lead to assigning low trust values to affected nodes that are otherwise trustworthy. Second, the continuous monitoring of link quality indicators is a significant energy overhead. For example, Su et al. [18] and Zhang et al. [28] used a triangle model for link quality assessment based on the packet received rate (PRR), the signal-to-noise ratio (SNR), and the LQI for assessing the quality of the link between two nodes: A and B. As the model was designed for underwater environments in which signal attenuation is high, it may not be generalizable to terrestrial WSNs. Second, the SNR and PRR calculations rely on real-time measurements, which may be delayed or may be prone to errors in noisy environments.

Further research may consider incorporating adaptive thresholds for the LQI or SNR based on historical data or on environmental factors to reduce reliance on static thresholds and to enhance model adaptability. The continuous monitoring of links could be replaced with periodic sampling to improve energy efficiency.

4.5. Authentication Delay

Most of the reviewed trust models imply the use of cryptography-based node authentication. However, the lengthy signature verification process may affect the calculation of the trust value [41]. To mitigate the risk, Lyu et al. [41] proposed the sharing of authentication information between neighbors rather than authenticating every received message. The approach aims to achieve a balance between the security and efficiency needs of the WSN. However, this approach assumes that neighboring nodes can be trusted to share accurate authentication data, which introduces potential vulnerabilities. For example, malicious nodes could manipulate this process by providing false authentication data to undermine trust evaluation or prioritize their own messages.

Future models could dynamically adjust authentication probabilities based on real-time network conditions. For instance, historical trust scores, packet loss rates, and environmental data could be used to predict high-risk nodes or events, enabling a smarter and more adaptive selective authentication approach.

4.6. Trust-Based Routing

While trust models can defend WSNs against DoS attacks that cause observable changes in node behavior, they are less effective in stopping attacks that lead to network resource overuse or communication breakdown. Several of the reviewed studies explored the use of trust information by routing protocols as a means of strengthening the defense of WSNs. For example, the trust model proposed by Rani et al. [42] works within a routing protocol. The cluster head that controls intra-cluster communication uses trust information to prevent communication with untrustworthy nodes. The trust evidence includes the number of successful or failed data transmissions between the cluster head and the sensors in the cluster or between the cluster head and other cluster heads. However, the model does not consider node authentication and the data/control packet forwarding metric, which limits the range of DoS attacks it can counter. Somewhat similarly, the routing protocol proposed in [41] uses trust information to select forwarding nodes and achieve better transmission efficiency. In addition, each node is assigned a verification probability that is dynamically adjusted, which makes the network more resistant to DoS attacks. However, the trust model focuses mainly on packet forwarding trust evidence. It ignores other direct trust evaluation metrics and does not consider indirect trust evaluation. Finally, the SDWSNs trust routing protocol proposed in [4] considers the trust values and the remaining energy of the nodes when selecting a data transmission path. However, the model is rather complex as it operates both in the control and the data planes of the network and involves multiple network layers (base stations, controllers, cluster heads, and other sensor nodes) and the network's functional layers (control plane and data plane). The trust model also needs further refinement.

The models reviewed show that trust-based routing in WSNs is a promising approach to mitigating the impact of DoS attacks in WSNs. However, the reviewed models are neither limited nor too complex to be implemented. In addition, they do not consider the critically important data accuracy trust evidence. Further research is needed to develop effective and feasible trust-based routing protocols for WSNs.

5. Conclusions

Based on the definition of DoS, this study comprehensively summarizes the DoS attacks in WSNs that can be addressed through trust models, including well-known DoS attacks as well as the new types of DoS attacks that have emerged with the development of WSNs. Furthermore, this study identifies the trust evidence needed to develop effective trust models for the detection of DoS attacks in WSNs and for the mitigation of their impact, and it compares trust evidence extraction methods and the trust evaluation methods.

The challenges identified in this study can be used as an evaluation framework for trust models that aim to protect WSNs against DoS attacks. As trust models consider node behavior manifestations rather than the cause of such behavior, the findings of this study are applicable as well to models that aim to maintain WSN functionality and reliability even when a node is faulty or has been compromised in an internal attack.

Further research needs to consider the three significant challenges that emerged: determining the threshold limits in trust evaluation, allocating trust metric weightings, and managing trust information loss. Design considerations to guide the development of future trust models include the impact of link quality on trust evaluation, the impact of authentication delay on trust evaluation, and the need to integrate trust models with routing protocols to actively defend against attacks rather than only focusing on detection.

Another direction for further research is the development of trust models for countering DoS attacks specific to the SDWSN domain. It is essential to ensure a high security standard when deploying an SDWSN architecture on a large scale and for geographically dispersed applications of WSNs.

This study has several limitations. First, our literature review is limited to studies published in English. Second, our review focuses solely on the research of trust models for dealing with DoS attacks in WSNs. It does not cover studies of trust models related to other security issues, such as privacy protection and access control, or other concerns, such as data aggregation and filtering. Third, to streamline our study, we excluded research involving AI and blockchains. These limitations can be overcome in further studies.

Author Contributions: Conceptualization, L.W., K.P. and M.L.Y.; methodology, L.W., K.P. and M.L.Y.; validation, K.P. and M.L.Y.; formal analysis, L.W.; investigation, L.W.; resources, M.L.Y.; data curation, L.W.; writing—original draft preparation, L.W., K.P. and M.L.Y.; writing—review and editing, L.W. and K.P.; visualization, L.W.; supervision, K.P. and M.L.Y.; project administration, K.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Summary of data extraction information.

No	Author, Year	Trust Evidence	Direct Trust	Indirect Trust	Evaluation Methods	Trust Metric Range	Distributed	Centralized	Hybrid	Detect Attacks	Defend Attacks	DoS Attack Types	Simulation Tool	Domain
1	Ganeriwal et al., 2004, 2008 [5,38]	Forwarding trust, data trust.	✓	✓	Bayesian beta method, aging factor, Dempster–Shafer belief theory and concept of belief discounting.	A realm number between 0 and 1	✓			✓	✓	Bad-Mouthing Attack, Good-Mouthing Attack, Sybil Attack.	NESLsim	WSN
2	Cao et al., 2006 [36]	Ratio of the number of successfully solved puzzles to the number of packets sent.	✓		Take the logarithm of the ratio.	Negative real numbers less than or equal to 0		✓		✓	✓	Flooding Attack.	Not mentioned	WSN
3	Cho and Qu, 2013 [37]	Forwarding trust.	✓		Bayesian beta method, entropy-based trust models.	A realm number between 0 and 1	✓			✓	✓	Selective Forwarding Attack.	OPNET	WSN
4	Han et al., 2014 [40]	Energy consumption, probability of packet reached successfully (PPRS).	✓		Bayesian beta method, aging factor.	A realm number between 0 and 1			✓	✓		Selective Forwarding Attack, Flooding Attack.	NS2	Cluster-based WSN

Table A1. *Cont.*

No	Author, Year	Trust Evidence	Direct Trust	Indirect Trust	Evaluation Methods	Trust Metric Range	Distributed	Centralized	Hybrid	Detect Attacks	Defend Attacks	DoS Attack Types	Simulation Tool	Domain
5	Gautam and Kumar, 2018 [39]	Probability of packet reached successfully (PPRS).	✓	✓	Bayesian beta method, forgetting curve.	A realm number between 0 and 1	✓			✓		Unclassified.	Not mentioned	Cluster-based WSN
6	Jinhui et al., 2018 [13]	Energy consumption, power consumption sequence.	✓		Bayesian beta method, penalty factor, threshold, Pearson correlation coefficient.	An unsigned integer between 0 and 10		✓		✓		Hybrid DoS Attacks.	NS2	Cluster-based WSN
7	Wang et al., 2018 [4]	Data forwarding trust, control forwarding trust, packet-in trust.	✓		Bayesian beta method, weight, threshold.	Local trust: an unsigned integer between 0 and 10 Global trust: a realm number between 0 and 1			✓	✓	✓	Selective Forwarding Attack, New-Flow Attack.	Contiki Cooja 2.7	SDWSN

Table A1. *Cont.*

No	Author, Year	Trust Evidence	Direct Trust	Indirect Trust	Evaluation Methods	Trust Metric Range	Distributed	Centralized	Hybrid	Detect Attacks	Defend Attacks	DoS Attack Types	Simulation Tool	Domain
8	Anwar et al., 2019 [35]	Packet Received Evaluation (PRE), Packet Sending Evaluation (PSE), Transit Packet Evaluation (TPE).	✓	✓	Bayesian beta method, weight.	A realm number between 0 and 1	✓			✓		On–Off Attack, Bad-Mouthing Attack, DoS Attack.	OMNET++	WSN
9	Usman et al., 2019 [7]	Data rate.	✓		Threshold.	Trust or not trust		✓		✓	✓	Unclassified.	MATLAB	WBAN
10	Lyu et al., 2019 [41]	Forwarding trust.	✓		Aging factor.	A realm number between 0 and 1	✓			✓	✓	Unclassified.	OPNET	IoT
11	Bin-Yahya and Shen, 2019 [45]	Forwarding trust, sending-rate trust, new-flow trust.	✓	✓	Bayesian beta method, weight, aging factor.	A realm number between 0 and 1			✓	✓		Blackhole Attack, Selective Forwarding Attack, DoS Attack.	MATLAB	SDWSN

Table A1. *Cont.*

No	Author, Year	Trust Evidence	Direct Trust	Indirect Trust	Evaluation Methods	Trust Metric Range	Distributed	Centralized	Hybrid	Detect Attacks	Defend Attacks	DoS Attack Types	Simulation Tool	Domain
12	Wu et al., 2019 [14]	Communication trust, data trust, energy trust.	✓	✓	Bayesian beta method, weight, threshold, aging factor, LQI analysis.	A realm number between 0 and 1	✓			✓		Selective Forwarding Attack, DoS Attack.	MATLAB	WSN
13	Qureshi et al., 2020 [15]	The number of sent packets, the number of received packets, the time of sending packets, the time of receiving packets, and the packet loss rate between two nodes.	✓	✓	Bayesian beta method, threshold.	A realm number between 0 and 1	✓			✓	✓	On and Off Attack, Bad-Mouthing Attack, DoS Attack.	OMNET++	IoT
14	Su et al., 2020 [18]	Communication trust, data trust, energy trust.	✓	✓	Bayesian beta method, weight, threshold, aging factor.	A realm number between 0 and 1	✓			✓		Selective Forwarding Attack, Flooding Attack.	MATLAB	UASN

Table A1. *Cont.*

No	Author, Year	Trust Evidence	Direct Trust	Indirect Trust	Evaluation Methods	Trust Metric Range	Distributed	Centralized	Hybrid	Detect Attacks	Defend Attacks	DoS Attack Types	Simulation Tool	Domain
15	Anand and Vasuki, 2021 [34]	Throughput, packet rate, packet forwarding rate, hop count, energy utilization.	✓	✓	An improved statistical method of the grading factor with the probability weight factor, Fleiss kappa function.	A realm number between 0 and 1	✓			✓		Selective Forwarding Attack, Flooding Attack.	NS-2.33	WSN
16	Rahamathullah and Karthikeyan, 2021 [24]	Data packet forwarding ratio, control packet forwarding ratio, energy consumption.	✓	✓	Weight, threshold.	A realm number between 0 and 1	✓			✓	✓	DDoS Attack.	Contiki Cooja 3.0, Tmote sky wireless sensor board	IoBT
17	Isaac Sajan and Jasper, 2021 [16]	The quantity of the packet delivered successfully, the previous history of packets dropped by the nodes, the similarity in attributes.	✓	✓	Weight, threshold.	A realm number between 0 and 1		✓		✓	✓	Vampire Attack.	MATLAB	Ad hoc sensor network (WANET)

Table A1. *Cont.*

No	Author, Year	Trust Evidence	Direct Trust	Indirect Trust	Evaluation Methods	Trust Metric Range	Distributed	Centralized	Hybrid	Detect Attacks	Defend Attacks	DoS Attack Types	Simulation Tool	Domain
18	Rani et al., 2021 [42]	Data trust, community trust.	✓	✓	Weight.	A realm number between 0 and 1	✓			✓	✓	Unclassified.	OMNET++	Cluster-based WSN
19	Yuvaraj et al., 2022 [43]	Forwarding trust, IMFs.	✓		Weight, DR-HHT, correlation coefficient, Kolmogorov–Smirnov test.	A realm number between 0 and 1	✓			✓	✓	LDoS.	MATLAB	IoT WSN
20	Bin-Yahya et al., 2022 [10]	Forwarding trust, sending rate trust, new-flow trust, node reliability.	✓	✓	Bayesian beta method, weight, threshold, aging factor, reward and penalize.	A realm number between 0 and 1			✓	✓		Blackhole Attack, Selective Forwarding Attack, DoS Attack, Good-Mouthing Attack, On–Off Attack, Hybrid DoS Attacks, New-Flow Attack.	MATLAB	SDWSN
21	Isong et al., 2023 [46]	Forwarded packets, received packets, traffic statistical information.	✓	✓	If drop or block packets value++, threshold, IDS module analysis.	An unsigned integer between 0 and 10			✓	✓	✓	Unclassified.	Not evaluated yet	SDWSN

Table A1. *Cont.*

No	Author, Year	Trust Evidence	Direct Trust	Indirect Trust	Evaluation Methods	Trust Metric Range	Distributed	Centralized	Hybrid	Detect Attacks	Defend Attacks	DoS Attack Types	Simulation Tool	Domain
22	Zhang et al., 2023 [28]	Communication trust, energy trust, data trust.	✓	✓	Bayesian beta method, weight, threshold, aging factor, reward and penalize, honesty degree.	A realm number between 0 and 1	✓			✓	✓	Bad-Mouthing Attack, Data Tampering Attack.	Not mentioned	UASN
23	Khan and Singh, 2023 [3]	Communication trust, data trust.	✓	✓	Bayesian beta method, weight, threshold, aging factor, reward and penalize	A realm number between 0 and 4			✓	✓	✓	Bad-Mouthing Attack, Blackhole Attack, Grey Hole Attack.	MATLAB	Cluster-based WSN
24	Iftikhar et al., 2023 [29]	Packet drop rate, penalized over transmission paths.	✓	✓	Bayesian beta model, weight, threshold, aging factor, penalize	A realm number between 0 and 1	✓			✓	✓	DoS Attack, DDoS Attack, On–Off Attack, Bad-Mouthing Attack, Sybil Attack.	OMNET++	Edge IoT
25	Ahmed et al., 2024 [17]	Communication trust, energy trust, data trust.	✓	✓	Bayesian beta method, weight, aging factor, penalizing factor, load balancing.	A realm number between 0 and 1	✓			✓	✓	Unclassified.	MATLAB	Edge IoT
26	Saidi, 2024 [2]	Communication trust, energy trust, data trust.	✓	✓	Bayesian beta method, threshold, aging factor.	A realm number between 0 and 1			✓	✓	✓	Data Tampering Attack, Blackhole Attack, Sinkhole Attack.	MATLAB	WSN

Table A1. *Cont.*

No	Author, Year	Trust Evidence	Direct Trust	Indirect Trust	Evaluation Methods	Trust Metric Range	Distributed	Centralized	Hybrid	Detect Attacks	Defend Attacks	DoS Attack Types	Simulation Tool	Domain
27	Singh et al., 2024 [31]	Encounter frequency, packet forwarding, recent contact, contact durability.	✓	✓	Bayesian beta method, logarithm of the ratio.	A realm number between 0 and 1	✓			✓	✓	Unclassified.	ONE	Opportunistic IoT
28	Ye and Jiang, 2024 [30]	Energy trust, historical interaction records, data trust.	✓	✓	Logarithm of the ratio, statistical method, aging factor.	A realm number between 0 and 1	✓			✓	✓	Blackhole Attack, DoS Attack, Grey Hole Attack.	MATLAB	UASN
29	Almutairi et al., 2024 [33]	Spatial similarity, temporal similarity, communication, delay, energy.	✓	✓	Bayesian beta method, penalizing factor, aging factor, weight.	A realm number between 0 and 1	✓			✓	✓	Physical Mobility Attack, Selfish Attack, Selective Forwarding Attacks, DoS Attack.	AquaSim-NG	Internet of Underwater Things
30	Han et al., 2024 [32]	Receiving and sending data packet.	✓	✓	Bayesian beta method, penalizing factor, weight, improved density peaks clustering algorithm.	A realm number between 0 and 1	✓			✓	✓	On–Off Attack, Bad-Mouthing Attack, Collusion Attacks.	MATLAB	WSN

References

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: A survey. *Comput. Netw.* **2002**, *38*, 393–422. [CrossRef]
2. Saidi, A. An adaptive trust system for misbehavior detection in wireless sensor networks. *Wirel. Netw.* **2024**, *30*, 2589–2615. [CrossRef]
3. Khan, T.; Singh, K. RTM: Realistic Weight-Based Reliable Trust Model for Large Scale WSNs. *Wirel. Pers. Commun.* **2023**, *129*, 953–991. [CrossRef]
4. Wang, R.; Zhang, Z.; Zhang, Z.; Jia, Z. ETMRM: An Energy-efficient Trust Management and Routing Mechanism for SDWSNs. *Comput. Netw.* **2018**, *139*, 119–135. [CrossRef]
5. Ganeriwal, S.; Balzano, L.K.; Srivastava, M.B. Reputation-based framework for high integrity sensor networks. *ACM Trans. Sens. Netw.* **2008**, *4*, 1–37. [CrossRef]
6. Wood, A.D.; Stankovic, J.A. Denial of service in sensor networks. *Computer* **2002**, *35*, 54–62. [CrossRef]
7. Usman, M.; Asghar, M.R.; Ansari, I.S.; Granelli, F.; Qaraqe, M. Trust-Based DoS Mitigation Technique for Medical Implants in Wireless Body Area Networks. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
8. Gungor, V.C.; Hancke, G.P. Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches. *IEEE Trans. Ind. Electron.* **2009**, *56*, 4258–4265. [CrossRef]
9. Yick, J.; Mukherjee, B.; Ghosal, D. Wireless sensor network survey. *Comput. Netw.* **2008**, *52*, 2292–2330. [CrossRef]
10. Bin-Yahya, M.; Alhussein, O.; Shen, X. Securing Software-Defined WSNs Communication via Trust Management. *IEEE Internet Things J.* **2022**, *9*, 22230–22245. [CrossRef]
11. Govindan, K.; Mohapatra, P. Trust computations and trust dynamics in mobile adhoc networks: A survey. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 279–298. [CrossRef]
12. Reddy, V.B.; Venkataraman, S.; Negi, A. Communication and data trust for wireless sensor networks using D–S theory. *IEEE Sens. J.* **2017**, *17*, 3921–3929. [CrossRef]
13. Jinhui, X.; Yang, T.; Feiyue, Y.; Leina, P.; Juan, X.; Yao, H. Intrusion Detection System for Hybrid DoS Attacks using Energy Trust in Wireless Sensor Networks. *Procedia Comput. Sci.* **2018**, *131*, 1188–1195. [CrossRef]
14. Wu, X.; Huang, J.; Ling, J.; Shu, L. BLTM: Beta and LQI Based Trust Model for Wireless Sensor Networks. *IEEE Access* **2019**, *7*, 43679–43690. [CrossRef]
15. Qureshi, K.N.; Iftikhar, A.; Bhatti, S.N.; Piccialli, F.; Giampaolo, F.; Jeon, G. Trust management and evaluation for edge intelligence in the Internet of Things. *Eng. Appl. Artif. Intell.* **2020**, *94*, 103756. [CrossRef]
16. Sajan, R.I.; Jasper, J. A secure routing scheme to mitigate attack in wireless adhoc sensor network. *Comput. Secur.* **2021**, *103*, 102197. [CrossRef]
17. Ahmed, A.; Qureshi, K.N.; Anwar, M.; Masud, F.; Imtiaz, J.; Jeon, G. Link-based penalized trust management scheme for preemptive measures to secure the edge-based internet of things networks. *Wirel. Netw.* **2024**, *30*, 4237–4259. [CrossRef]
18. Su, Y.; Mal, S.; Jin, Z.; Fu, X.; Li, Y.; Liu, X. A Trust Model for Underwater Acoustic Sensor Networks Based on Fast Link Quality Assessment. In Proceedings of the Global Oceans 2020: Singapore—U.S. Gulf Coast, Biloxi, MS, USA, 5–30 October 2020; pp. 1–6.
19. Muzammal, S.M.; Murugesan, R.K.; Jhanjhi, N.Z. A Comprehensive Review on Secure Routing in Internet of Things: Mitigation Methods and Trust-Based Approaches. *IEEE Internet Things J.* **2021**, *8*, 4186–4210. [CrossRef]
20. Alhandi, S.A.; Kamaludin, H.; Alduais, N.A.M. Trust Evaluation Model in IoT Environment: A Comprehensive Survey. *IEEE Access* **2023**, *11*, 11165–11182. [CrossRef]
21. Tyagi, H.; Kumar, R.; Pandey, S.K. A detailed study on trust management techniques for security and privacy in IoT: Challenges, trends, and research directions. *High-Confid. Comput.* **2023**, *3*, 100127. [CrossRef]
22. Zhu, R.; Boukerche, A.; Long, L.; Yang, Q. Design Guidelines On Trust Management for Underwater Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2024**, *26*, 2547–2576. [CrossRef]
23. Chandrasekaran, S.K.; Rajasekaran, V.A. Trust evaluation model in IoT environment: A review. *Environ. Dev. Sustain.* **2024**, 1–32. [CrossRef]
24. Rahamathullah, U.; Karthikeyan, E. A lightweight trust-based system to ensure security on the Internet of Battlefield Things (IoBT) environment. *Int. J. Syst. Assur. Eng. Manag.* **2021**. [CrossRef]
25. Kitchenham, B.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*; Keele University and University of Durham: Durham, UK, 2007.
26. Bukar, U.A.; Othman, M. Architectural Design, Improvement, and Challenges of Distributed Software-Defined Wireless Sensor Networks. *Wirel. Pers. Commun.* **2022**, *122*, 2395–2439. [CrossRef]
27. Ouzzani, M.; Hammady, H.; Fedorowicz, Z.; Elmagarmid, A. Rayyan—A web and mobile app for systematic reviews. *Syst. Rev.* **2016**, *5*, 210. [CrossRef] [PubMed]

28. Zhang, M.; Feng, R.; Zhang, H.; Su, Y. A recommendation management defense mechanism based on trust model in underwater acoustic sensor networks. *Future Gener. Comput. Syst.* **2023**, *145*, 466–477. [CrossRef]
29. Iftikhar, A.; Qureshi, K.N.; Altalbe, A.A.; Javeed, K. Security Provision by Using Detection and Prevention Methods to Ensure Trust in Edge-Based Smart City Networks. *IEEE Access* **2023**, *11*, 137529–137547. [CrossRef]
30. Ye, J.; Jiang, W. Routing Protocol for Underwater Wireless Sensor Networks Based on a Trust Model and Void-Avoided Algorithm. *Sensors* **2024**, *23*, 7614. [CrossRef]
31. Singh, J.; Dhurandher, S.K.; Woungang, I.; Chao, H.-C. Context-Aware Trust and Reputation Routing Protocol for Opportunistic IoT Networks. *Sensors* **2024**, *24*, 7650. [CrossRef]
32. Han, Y.; Wang, H.; Li, Y.; Zhang, L. Trust-aware and improved density peaks clustering algorithm for fast and secure models in wireless sensor networks. *Pervasive Mob. Comput.* **2024**, *105*, 101993. [CrossRef]
33. Almutairi, A.; Carpent, X.; Furnell, S. Towards a Mobility-Aware Trust Model for the Internet of Underwater Things. In Proceedings of the ICT Systems Security and Privacy Protection, Edinburgh, UK, 26 July 2024; pp. 1–15.
34. Anand, C.; Vasuki, N. Trust Based DoS Attack Detection in Wireless Sensor Networks for Reliable Data Transmission. *Wirel. Pers. Commun.* **2021**, *121*, 2911–2926. [CrossRef]
35. Anwar, R.W.; Zainal, A.; Outay, F.; Yasar, A.; Iqbal, S. BTEM: Belief based trust evaluation mechanism for Wireless Sensor Networks. *Future Gener. Comput. Syst.* **2019**, *96*, 605–616. [CrossRef]
36. Cao, Z.; Zhou, X.; Xu, M.; Chen, Z.; Hu, J.; Tang, L. Enhancing Base Station Security Against DoS Attacks in Wireless Sensor Networks. In Proceedings of the 2006 International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, China, 22–24 September 2006; pp. 1–4.
37. Cho, Y.; Qu, G. Detection and Prevention of Selective Forwarding-Based Denial-of-Service Attacks in WSNs. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 205920. [CrossRef]
38. Ganeriwal, S.; Srivastava, M.B. Reputation-based framework for high integrity sensor networks. In Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, Washington, DC, USA, 25 October 2004; pp. 66–77.
39. Gautam, A.K.; Kumar, R. A Robust Trust Model for Wireless Sensor Networks. In Proceedings of the 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Gorakhpur, India, 2–4 November 2018; pp. 1–5.
40. Han, G.; Shen, W.; Duong, T.Q.; Guizani, M.; Hara, T. A proposed security scheme against Denial of Service attacks in cluster-based wireless sensor networks. *Secur. Commun. Netw.* **2014**, *7*, 2542–2554. [CrossRef]
41. Lyu, C.; Zhang, X.; Liu, Z.; Chi, C.H. Selective Authentication Based Geographic Opportunistic Routing in Wireless Sensor Networks for Internet of Things Against DoS Attacks. *IEEE Access* **2019**, *7*, 31068–31082. [CrossRef]
42. Rani, P.; Gupta, N.K. Composite Trust for Secure Routing Strategy through Energy based Clustering in WSN. In Proceedings of the 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 19–20 February 2021; pp. 1–6.
43. Yuvaraj, D.; Priya, S.-S.; Braveen, M.; Krishnan, S.-N.; Nachiyappan, S.; Mehbodniya, A.; Ahamed, A.-M.-U.; Sivaram, M. Novel DoS Attack Detection Based on Trust Mode Authentication for IoT. *Intell. Autom. Soft Comput.* **2022**, *34*, 1505–1522. [CrossRef]
44. Miyazaki, T.; Yamaguchi, S.; Kobayashi, K.; Kitamichi, J.; Song, G.; Tsukahara, T.; Hayashi, T. A software defined wireless sensor network. In Proceedings of the 2014 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 3–6 February 2014; pp. 847–852.
45. Bin-Yahya, M.; Shen, X. HTM: Hierarchical Trust Management for Software-Defined WSNs. In Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
46. Isong, B.; Manuel, M.; Dladlu, N.; Abu-Mahfouz, A. Trust Management Framework for Securing Software-Defined Wireless Sensor Networks. In Proceedings of the 2023 International Conference on Electrical, Computer and Energy Technologies (ICECET), Cape Town, South Africa, 16–17 November 2023; pp. 1–6.
47. Webber, J.L.; Arafa, A.; Mehbodniya, A.; Karupusamy, S.; Shah, B.; Dahiya, A.K.; Kanani, P. An efficient intrusion detection framework for mitigating blackhole and sinkhole attacks in healthcare wireless sensor networks. *Comput. Electr. Eng.* **2023**, *111*, 108964. [CrossRef]