International Journal of Safety and Security Engineering
Vol. 16, No. 9, September, 2026, pp. 2143-2152
Journal homepage: http://iieta.org/journals/ijsse

TBAR-IoT: A Lightweight Trust-Based Adaptive Routing Protocol for Secure IoT Networks under Blackhole Attack

Bhaskar Maithani[1*], Ajay Vikram Singh[1], Said Broumi[2,3]

[1] Amity Institute of Information Technology, Amity University, Noida 201313, India
[2] Laboratory of Information Processing, Faculty of Science Ben M'Sik, Hassan II University of Casablanca, Casablanca 20700, Morocco
[3] Department of Computer Science, Regional Center for the Professions of Education and Training (C.R.M.E.F), Casablanca 20102, Morocco

Corresponding Author Email: bhaskarmaithani021@gmail.com

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/ijsse.160917

Received: 18 August 2026
Revised: 10 September 2026
Accepted: 20 September 2026
Available online: 30 September 2026

Keywords:
adaptive routing, blackhole attack, Internet of Things, lightweight trust model, OMNeT++ simulation, trust management

ABSTRACT

Due to its dynamic and resource-constrained nature, secure routing in the Internet of Things (IoT) is a major challenge, and the routing process is vulnerable to attack. Among such attacks, the blackhole attack is one of the most common, where malicious nodes advertise manipulated routing information to trap and drop packets. Existing approaches fail to mitigate blackhole attacks while maintaining energy efficiency. To address this, we propose Trust-Based Adaptive Routing for IoT (TBAR-IoT), a lightweight, trust-based protocol using an adaptive multi-parameter trust model to evaluate trust at runtime. The protocol fuses behavioral and contextual features, including Packet Forwarding Ratio (PFR), communication frequency, recency, and residual energy, into a single trust value used to select reliable, energy-efficient nodes, with deliberate packet dropping distinguished from loss caused by link failure, mobility, or congestion before a node is isolated. We implemented TBAR-IoT in OMNeT++ 6.0 and evaluated it against ad hoc on-demand distance vector (AODV), Lightweight and Dependable Trust System (LDTS), and Trust-based Ad hoc On-demand Multipath Distance Vector (Trust-AOMDV) under blackhole attack. TBAR-IoT achieved a packet delivery ratio (PDR) of 91.2%, compared to 61.8%, 73.1%, and 81.0% for AODV, LDTS, and Trust-AOMDV, respectively, while at the same time recording the lowest energy consumption per delivered packet and the lowest average end-to-end delay of the four protocols. The protocol tolerates up to 20% malicious nodes while maintaining PDR above 85%, making it well suited for adaptive IoT deployments under blackhole attack.

1. INTRODUCTION

The Internet of Things (IoT) has emerged as one of the most disruptive computing paradigms of the new millennium, enabling the interconnection of billions of heterogeneous devices to realize innovative environments across healthcare, industrial automation, agriculture, and smart cities [1-3]. These devices operate autonomously in self-organized, multi-hop wireless networks where sensing, computation, and communication occur under strict constraints of energy, memory, and processing capability. Despite these advantages, IoT networks face significant challenges in achieving secure and reliable routing due to their decentralized architecture, node mobility, and absence of centralized supervision [4-6].

Conventional routing protocols such as ad hoc on-demand distance vector (AODV) and dynamic source routing (DSR) were initially designed for cooperative mobile ad hoc networks (MANETs), assuming honest participation by all nodes [7]. These protocols mainly depend on hop count or shortest-path metrics and ignore node reliability, residual energy, and behavioral stability [8]. Although this assumption simplifies protocol design, it also leaves such protocols highly vulnerable to attack in adversarial IoT environments. Specifically, one of the most serious adverse effects on the delivery ratios of packets, latency, and energy usage is a blackhole attack, in which malicious nodes advertise a fabricated optimal path and intentionally drop packets [9-11]. Conventional routing algorithms are therefore unsuitable for large-scale, unprotected IoT deployments. Routing in IoT differs substantially from that in traditional ad hoc networks: the networks are larger and more heterogeneous, their topology changes continuously, and nodes are deployed in uncontrolled environments where unprotected wireless channels and the absence of central authentication can be exploited by such attacks. Using hop count alone may select unreliable or energy-depleted nodes, which leads to packet loss and reduced network lifetime, so routing in IoT needs to be context-aware, accounting for node behavior, residual energy, and reliability as well as path length.

These deficiencies have motivated the development of trust-aware routing algorithms, in which nodes assess the trustworthiness of their neighbors from observed behavior

[12-14]. Trust is a quantifiable expectation that a node will behave as prescribed under conditions of uncertainty, usually expressed as a normalized value between zero and one that changes over time through cumulative interaction; because it is local, non-transitive, and context-specific, it is well suited to the decentralized IoT, where no central authority exists [12]. Trust management also complements cryptographic protection: cryptographic methods can authenticate node identities but do not guarantee honest packet forwarding, so insider threats and behavioral anomalies must be countered by evaluating node behavior during network operation. Trust-aware routing is more flexible because it allows the network to favor trusted nodes and isolate malicious ones. A prominent solution in this area is the Lightweight and Dependable Trust System (LDTS) proposed by Li et al. [15], which aggregates trust hierarchically by combining direct and indirect observations, thereby reducing communication overhead and improving scalability. However, LDTS depends on stable clustering, which limits its applicability in highly dynamic IoT environments, and its emphasis on trust aggregation rather than routing decisions reduces its effectiveness against blackhole attacks.

Recent research has proposed lightweight, behavior-based trust mechanisms to address these issues [16]. For example, LETM-IoT, proposed by Hassan et al. [17], performs efficient trust computation with low overhead and detects Sybil and behavioral attacks. Such approaches are effective, but they focus primarily on detection accuracy and do not fully integrate trust into adaptive routing decisions. Machine-learning intrusion detection has also been applied to the RPL routing stack for low-power IoT networks [18], and reinforcement learning has been combined with runtime trust evaluation to adapt routing to changing node behavior in vehicular networks [19]; both, however, carry computational requirements beyond the budget of resource-constrained IoT devices. Likewise, Trust-based Ad hoc On-demand Multipath Distance Vector (Trust-AOMDV), introduced by Govindaraj and Arumugam [20], adds behavioral trust to multipath routing to improve resilience against blackhole and selective forwarding attacks. However, multipath trust-based designs consume more energy and require greater control overhead, which limits their applicability to dense, low-resource IoT networks.

The literature indicates that trust must be measured dynamically and along multiple dimensions. Sagar et al. [21] noted that metrics such as Packet Forwarding Ratio (PFR), residual energy, interaction recency, and recommendation credibility are essential for trust management in IoT. Similarly, Ahmadi and Javidan [22] showed that insider attacks such as blackhole and selective packet dropping can be detected by hybrid trust-anomaly detection models, and the need to separate deliberate dropping from unintentional loss when evaluating forwarding behavior has been identified as central to trust-based mitigation of packet-dropping attacks [23]. Work published since has continued along the same lines: adaptive trust combined with anomaly detection for IoT routing [24], lightweight trust modeling for constrained devices [25], AI-assisted trust and decision modeling in software-defined IoT deployments [26], and cross-layer defense against blackhole and wormhole attacks in ad hoc networks [27]. A recent review of cyber-attacks targeting IoT systems and their countermeasures surveys this landscape [28]. Despite these developments, most studies rely on simplified simulations or restricted attack conditions, which limits their relevance to real-world IoT deployments [22].

Table 1 summarizes the characteristics of these schemes. Across them, three major limitations recur:

(i) Reliance on static, single-metric trust computation that does not adapt to changing network conditions;

(ii) High computational or communication cost, which makes them impractical for low-power IoT devices; and

(iii) Dependence on additional structure — cluster formation, multipath maintenance, a separate detection stage, or a particular routing stack — which constrains where the scheme can be deployed, and evaluation that is largely confined to simplified or non-aggressive attack models.

Table 1. Comparison of representative trust-based routing and detection schemes

Scheme	Trust Basis	Key Limitation
LDTS [15]	Cluster-based trust aggregation	Requires stable clustering
Trust-AOMDV [20]	Multipath behavioral trust	High control overhead
LETM-IoT [17]	Lightweight behavioral trust	Detection only, not routing
MCTE-RPL [16]	Multi-context trust in RPL	Bound to the RPL stack
ML-based RPL IDS [18]	Learned intrusion classifier	Training cost; RPL-specific
Adaptive trust IDS [24]	Trust with anomaly scoring	Detection stage separate from routing

Note: LDTS = Lightweight and Dependable Trust System; AOMDV = Ad Hoc On-Demand Multipath Distance Vector; IoT = Internet of Things; RPL = Routing Protocol for Low-Power and Lossy Networks; ML = Machine Learning; IDS = Intrusion Detection System.

To address these gaps, this paper proposes an adaptive trust-based routing framework, the Trust-Based Adaptive Routing for IoT (TBAR-IoT) protocol. TBAR-IoT uses an adaptive multi-parameter behavioral trust model to inform routing decisions. Within a flat, single-path trust scheme, TBAR-IoT computes a single trust value locally by aggregating behavioral attributes, namely interaction recency, frequency, duration, PFR, and residual energy. Trust values are updated at every interaction so that the protocol adapts quickly to network dynamics and identifies malicious nodes at an early stage. The resulting trust update mechanism is lightweight and decentralized, which substantially lowers computational cost while preserving resistance to routing attacks.

2. MATERIALS AND METHODS

2.1 Proposed work

To mitigate blackhole attacks, we present the TBAR-IoT protocol, designed to address the challenges typical of IoT ecosystems, namely dynamically changing topology, constrained power resources, and heterogeneous routing behavior. TBAR-IoT uses a lightweight multiplicative trust model that estimates node reliability dynamically from five locally observed factors, namely interaction recency, interaction frequency, interaction duration, residual energy and PFR, in order to identify trustworthy nodes. The major innovations provided by TBAR-IoT are:

(1) Dynamic Performance Optimization: TBAR-IoT continuously adapts to network conditions by choosing

reliable nodes based on dynamic trust measures such as forwarding reliability and node activity characteristics.

(2) Lightweight Multiplicative Trust Model: TBAR-IoT uses a simplified trust computation scheme based on the weighted geometric mean of five behavioral parameters with a fixed-weight threshold activation. The weights are fixed by design rather than learned, since on-node training is not feasible under the memory and energy budgets of constrained IoT devices; no claim of learned behavior is made.

(3) Advantage over Conventional Protocols: TBAR-IoT outperforms legacy AODV, LDTS, and Trust-AOMDV in terms of packet delivery ratio (PDR), energy consumption per delivered packet, and robustness.

(4) Explicit Blackhole Detection and Mitigation: a deliberate drop is distinguished from loss caused by collision, mobility, link failure or congestion by comparing what a node observes with what its other neighbors observe, and a confirmed attacker is excluded both locally and by its neighbors for a bounded hold-down period after which it may recover.

2.2 Weighted multi-parameter trust model

The reliability of nodes in dynamic IoT networks varies over time due to frequent topology reconfiguration, device heterogeneity, and random node failures. TBAR-IoT employs a weighted multi-parameter trust computation mechanism that processes real-time behavioral and environmental signals.

2.2.1 Mapping trust metrics in Internet of Things networks

TBAR-IoT computes the trust value of each node using five key parameters: interaction recency, communication frequency, interaction duration, residual energy, and PFR. The mapping is summarized in Table 2.

Table 2. Trust and routing parameters in Trust-Based Adaptive Routing for IoT (TBAR-IoT)

Actual Parameter	Mapped IoT Parameter
Time Since Last Interaction	P_1 (Recency)
Frequency of Communication	P_2 (Frequency)
Total Duration of Interaction	P_3 (Duration)
Residual Energy Level	P_4 (Energy Level)
Packet Forwarding Ratio (PFR)	P_5 (PFR)

Each neighbor is entered at the minimum trust value, $T(n_i, n_j) = 0$, and must earn trust through observed forwarding behavior; the value is then updated dynamically as the neighbor participates in forwarding. The trust update process is triggered whenever a new communication event occurs or the topology changes. Because every neighbor starts from the same value, selection at network start is resolved by the energy and recency tie-breaks of Algorithm 2 rather than by trust.

2.2.2 Trust computation

Trust in TBAR-IoT is directional: node n_i computes its own trust in each neighbor n_j from its own observations, and routine operation exchanges no trust value between nodes. The neighbors that n_i and n_j have in common are asked for their own observations only after a node has already failed the test of Eq. (3), as specified in Algorithm 3. The evaluation rests on a vector of five behavioral observations that n_i holds about n_j, as shown in Eq. (1):

$$f(n_i, n_j) = (P_1, P_2, P_3, P_4, P_5) \tag{1}$$

where, P_1 is the interaction recency, P_2 the interaction frequency, P_3 the interaction duration, P_4 the residual energy level and P_5 the PFR, each normalized to the unit interval as defined below; $P_i(n_i, n_j)$ denotes the value of P_i that n_i observes for n_j.

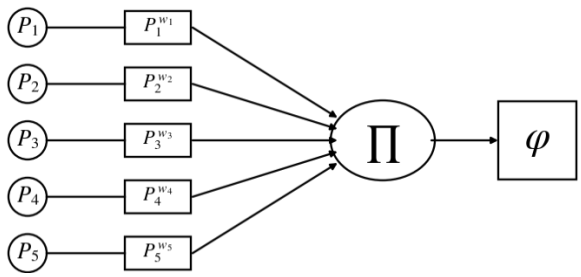


Figure 1. Threshold activation function with trust parameters

In Figure 1, the parameters P_1, P_2, P_3, P_4, P_5 carry the weights w_1, w_2, w_3, w_4, w_5 as exponents and are combined at a product node before the threshold activation. Here, P_i denotes the normalized trust parameters and w_i the weight of each parameter. The individual trust parameters are defined as follows: $P_1 = 1 - min(\Delta t / T_w, 1)$, where Δt is the time elapsed since the last interaction with the neighbor and T_w is the observation window, so that $P_1 = 1$ for an interaction in progress and falls linearly to 0 once a full window has passed without contact; P_2 = Frequency of interactions from n_i to n_j, where $i \neq j$ and $i, j \in \{1, 2, ..., N\}$, N being the number of nodes in the network, normalized as P_2/P_{2max} where P_{2max} is the highest observed frequency among neighbors; P_3 = Total duration of interactions, normalized as P_3/T_w over the same observation window T_w; P_4 = Residual energy level, normalized as $E_{remaining}/E_{initial}$; and P_5 = PFR. The corresponding weights are equal, $w_1 = w_2 = w_3 = w_4 = w_5 = 0.2$, so that their sum is unity. For each node n_i, this function activates for all neighboring nodes, i.e., for all n_1, n_2, ..., n_{i-1} and n_{i+1}, n_{i+2}, ..., n_N, as shown in Eq. (2):

$$x = \prod_{i=1}^{5} P_i^{w_i} \tag{2}$$

This value is passed through the threshold activation function φ, as shown in Eq. (3):

$$Z = \varphi(x) \tag{3}$$

where, the activation function is defined as shown in Eq. (4):

$$Z = \begin{cases} 0, & \text{if } x < \theta \text{ (node untrusted)} \\ 1, & \text{if } x \geq \theta \text{ (node trusted)} \end{cases} \tag{4}$$

The stored trust value is then moved towards the observed evidence x by a fixed fraction of the difference between them, as given in Eq. (5):

$$T_{new}(n_i, n_j) = T_{old}(n_i, n_j) + \eta[x - T_{old}(n_i, n_j)], \quad 0 < \eta < 1 \tag{5}$$

The five parameters are weighted equally, $w_1 = w_2 = w_3 = w_4 = w_5 = 0.2$. Equal weighting is a deliberate design decision rather than a fitted result: the model provides no principled

basis on which to rank one observation above another, so the symmetric assignment is adopted and no parameter-specific weight has to be estimated, tuned or defended. With all weights equal, Eq. (2) reduces to the unweighted geometric mean of the five parameters. The discriminating power of the scheme therefore comes from the form of the aggregation rather than from the choice of weights, which is what the following four properties establish. Aggregation is multiplicative rather than additive, and this choice gives the trust function four properties that a weighted sum does not possess. First, since every P_i lies in [0, 1] and the weights sum to unity, the aggregate score satisfies $x \in [0, 1]$; the activation threshold can therefore be placed in the interior of the range and is set to the midpoint, $\theta = 0.5$, a value attainable from both directions. Second, x is strictly increasing in every P_i whenever the remaining parameters are positive, so for each i there exist parameter vectors at which a change in P_i alone changes the activation outcome; no parameter is ignored by the decision. Third, the aggregation is non-compensatory: $x = 0$ if and only if the smallest of the five parameters is 0, so a node that forwards no packets receives an aggregate score of zero however favorable its contextual indicators appear. The same property applies in the opposite direction, which is why the boundary conditions stated below exempt a neighbor with no observation of some parameter from evaluation rather than scoring it at zero. A weighted sum offers no such guarantee, because a node scoring highly on four parameters can offset a zero on the fifth. Fourth, by the inequality of arithmetic and geometric means the multiplicative score never exceeds the corresponding weighted sum, with equality only when all five parameters are equal, so the scheme is uniformly more conservative than additive aggregation. Packet dropping consequently requires no separate bias term and no externally signalled drop event: a node that drops forwarded traffic exhibits a falling P_5, which lowers x directly; the observation window over which P_5 is measured and the procedure that separates deliberate dropping from benign loss are specified later in this section and in the Nomenclature. P_1 decreases with the time elapsed since the last interaction, so a higher P_1 indicates more recent contact and therefore more current evidence. For implementation the test $x \geq \theta$ is equivalent to testing whether the product of the five parameters is at least 0.03125, the fifth power of θ, so no root extraction is required and the decision costs four multiplications and one comparison per neighbor.

The update rule of Eq. (5) gives the stored trust value four properties that the original increment did not possess. Since T_{new} is a convex combination of T_{old} and x with $0 < \eta < 1$, and both T_{old} and x lie in [0, 1], the updated value also lies in [0, 1]; the range declared in the Nomenclature is therefore respected by construction and no separate normalization or upper bound is required. The update is signed: when the observed evidence exceeds the stored value the trust rises, and when it falls below the stored value the trust falls, so misbehavior is penalized by the same rule that rewards cooperation and no additional penalty term is needed. Because the correction is proportional to the difference, the stored value tracks the current evidence rather than remaining at a historical value; the influence of any past observation is reduced by a further factor of $(1 - \eta)$ at each subsequent update, so the metric forgets past behavior geometrically as new observations arrive. Finally, a node whose behavior improves sees x rise again and its trust recovers along the same trajectory, so exclusion is not permanent. For example, a node holding $T = 0.75$ whose forwarding degrades so that x falls to 0.55, which is still above θ and therefore still admissible, is updated to $0.75 + 0.5(0.55 - 0.75) = 0.65$ at the next observation, then to 0.60 and 0.575, converging towards 0.55, so that it is progressively deprioritized by Algorithm 2 without being excluded. If it resumes forwarding and x returns to 0.80, the value climbs back from 0.575 through 0.6875 and 0.7438 towards 0.80. A node whose score falls below θ is not smoothed down gradually in this way: Eq. (3) marks it untrusted at once and Algorithm 3 is invoked on it. The update rate is set to $\eta = 0.5$, which weights the stored history and the current observation equally. The resulting trust value is stored in the Route Cache (RC) to guide routing decisions in the IoT network.

Two boundary conditions complete the definition. First, each normalized parameter is clipped to [0, 1], so that an interaction duration exceeding the observation interval, or a frequency equal to the observed maximum, cannot carry the score outside the unit interval. Second, because the aggregation is non-compensatory, a neighbor for which no observation of some parameter yet exists within the current window would otherwise receive $x = 0$ merely for being new. The activation of Eq. (3) is therefore evaluated only once at least one observation of every parameter has been recorded within T_w; until then the neighbor is marked unevaluated, is not subject to Algorithm 3, and is ordered by the energy and recency tie-breaks of Algorithm 2.

The two quantities play distinct roles. The aggregate score x of Eq. (2) is the instantaneous evidence and, through Eq. (3), governs admission: a neighbor whose current behavior places it below θ is not used as a next hop. The stored value T of Eq. (5) is the smoothed history and governs preference: among admitted neighbors, Algorithm 2 selects the one holding the highest T. Admission therefore responds immediately to a change in behavior, while the ranking remains stable against a single unrepresentative observation.

For example, suppose node n_1 observes $P_1 = 0.9$, $P_2 = 0.8$, $P_3 = 0.6$, $P_4 = 0.8$ and $P_5 = 0.95$ for neighbor n_2. The product of the five parameters is $0.9 \times 0.8 \times 0.6 \times 0.8 \times 0.95 = 0.3283$, and x is its fifth root, $x = 0.80 \geq \theta$, so $Z = 1$ and n_2 is accepted. Consider instead a neighbor n_3 that advertises attractive routes and therefore presents favorable contextual parameters, $P_1 = 0.9$, $P_2 = 0.9$, $P_3 = 0.6$ and $P_4 = 0.95$, but forwards almost nothing, $P_5 = 0.05$. The same computation gives a product of 0.02309 and $x = 0.47 < \theta$, so Z = 0 and n_3 is rejected. Note that n_3 is excluded despite matching or exceeding n_2 on all four contextual parameters, and that no packet-drop event has been signalled to the trust function: the activation outcome is determined by the trust parameters themselves. In the limiting case of a node that forwards nothing at all, $P_5 = 0$ gives $x = 0$, so rejection is immediate and independent of the remaining four parameters.

A falling PFR is a symptom rather than a diagnosis: a node may fail to forward because it is dropping traffic deliberately, but equally because of a collision, because the link broke, because it moved out of transmission range, or because its queue overflowed under load. TBAR-IoT separates these causes with three questions put to W, the neighbors that n_i and n_j have in common, and answers all three from the forwarding observations the trust model already records for P_5. The first asks whether n_j is receiving at all: if any member of W holds too few observations of n_j to judge it, its packets are not reaching n_j, so the loss lies on the link rather than in n_j and is attributed to collision, mobility or link failure. The second asks

whether n_j forwards for anyone: a node that still forwards for at least one member of W is not dropping indiscriminately and is left unclassified. The third asks whether the neighborhood itself is working: if the members of W are not forwarding for one another either, the loss is attributed to congestion or overload, so a benign but heavily loaded node is not classified as malicious. Only a node that receives packets and forwards for no one while its neighborhood forwards normally is classified as a blackhole, and it is then excluded both by n_i and by the neighbors n_i informs. The members of W are asked once, at the point where a node has already failed the test of Eq. (3), and never periodically, so the exchange adds no standing overhead. The procedure is stated in Algorithm 3 and its parameters are listed in the Nomenclature. Every exit path other than the final classification leaves the neighbor unclassified, so the procedure fails towards inaction rather than towards a false accusation.

2.2.3 Trust evaluation, routing and blackhole detection algorithms

The notations and parameter values used in this section are summarized in the Nomenclature at the end of the paper. Building on the trust model described above, TBAR-IoT operates through three procedures executed at each node n_i. Algorithm 1 evaluates and updates the trust value of every neighbor n_j from the five observed parameters using Eqs. (2)-(5). Algorithm 2 uses these trust values to select the next intermediate node on a route, resolving ties by residual energy and then by interaction recency. Algorithm 3 is invoked whenever Algorithm 1 marks a neighbor as untrusted; it separates deliberate packet dropping from losses caused by collisions, mobility, link failure or congestion before a node is classified as a blackhole and blacklisted.

Algorithm 1: Trust evaluation and update at node n_i

Steps:
1. if n_j is blacklisted and T_h has elapsed do
2. Remove n_j from the blacklist, clear its observation record and mark n_j unevaluated
3. else if n_j is blacklisted do
4. Skip n_j
5. end if
6. if n_j has just entered transmission range do
7. Create an RC entry with $T(n_i, n_j) \leftarrow 0$ and mark n_j unevaluated
8. end if
9. Read P_1, P_2, P_3, P_4, P_5 of n_j over T_w from the Trust Table
10. If any P_i has no observation within T_w, mark n_j unevaluated and move to the next neighbor
11. Compute x using Eq. (2) and $Z = \varphi(x)$ using Eq. (4)
12. Read $T_{old}(n_i, n_j)$ from RC; if no entry exists, set $T_{old}(n_i, n_j) \leftarrow 0$
13. Set $T_{new}(n_i, n_j) \leftarrow T_{old}(n_i, n_j) + \eta\,[x - T_{old}(n_i, n_j)]$ and store it in RC
14. if $Z = 0$ do
15. Mark n_j untrusted and invoke Algorithm 3 on n_j
16. else
17. Remove the untrusted mark from n_j
18. end if
19. Repeat for every neighbor of n_i

Note: Precondition: n_i holds an RC entry towards the destination D of a route from the source S, and a stored trust value for each of its neighbors.

Algorithm 2: Trust-based adaptive routing for intermediate nodes

Steps:
1. Take from RC the route on which n_i is an intermediate node between the source S and the destination D
2. Select as C the neighbors of n_i that lie on a route towards D and are not blacklisted
3. Send one packet per window to each unevaluated or untrusted member of C, so that it can be evaluated
4. Select as M the members of C that are neither untrusted nor unevaluated
5. if M is empty do
6. Send a route error to S, start a new route discovery and exit
7. end if
8. Reduce M to the members with the highest trust value
9. if M has more than one member do
10. Reduce M to the members with the highest residual energy E
11. end if
12. if M has more than one member do
13. Reduce M to the members with the highest P_1
14. end if
15. if M has more than one member do
16. Reduce M to one member chosen at random
17. end if
18. Send the packet to the single member of M as the next intermediate node
19. Repeat at every intermediate node until the destination is reached

Note: Precondition: n_j has been marked untrusted by Eq. (3) at node n_i. The other neighbors are asked only at this point and never periodically.

Algorithm 3: Blackhole detection and mitigation

Steps:
1. Take as n_j the neighbor of n_i that Eq. (3) has just marked untrusted
2. if $P_5(n_i, n_j)$ is supported by fewer than A_{min} samples do
3. Leave n_j unclassified and exit
4. end if
5. Select as W the other neighbors of n_i that are also neighbors of n_j, and request from each member its forwarding ratio for n_j and its forwarding ratio for the other members of W; if W is empty, leave n_j unclassified and exit
6. Check whether n_j receives packets from every member of W, that is whether each $n_k \in W$ holds at least A_{min} samples for n_j
7. if any $n_k \in$ W holds fewer than A_{min} samples for n_j do
8. An external event such as collision, mobility, link failure or overflow exists; leave n_j unclassified and exit
9. end if
10. Check whether n_j forwards packets to the members of W, that is $P_5(n_k, n_j) > 0$ for any $n_k \in W$
11. if $P_5(n_k, n_j) > 0$ for any $n_k \in W$ do
12. n_j forwards for at least one neighbor; leave n_j unclassified and exit
13. end if
14. Check whether the members of W forward packets for one another, that is whether every $n_k \in W$ reports $P_5 = 0$ for the other members of W
15. if every $n_k \in$ W reports $P_5 = 0$ for the other members of W do
16. The whole neighborhood is not forwarding; attribute

the loss to congestion or overload, leave n_j unclassified and exit
17. end if
18. Classify n_j as a blackhole node, since $P_5(n_k, n_j) = 0 \; \forall \; n_k \in W$ while at least one member of W is still forwarding normally: n_j receives packets from its neighbors but forwards for none of them
19. Set $T(n_i, n_j) \leftarrow 0$ and blacklist n_j at n_i for the hold-down period T_h
20. Purge from RC every route whose next hop is n_j and discard its route replies until T_h expires
21. Notify the neighbors of n_i so that n_j is excluded from their route selection as well
22.After T_h, Algorithm 1 clears the blacklist entry and re-admits n_j at $T(n_i, n_j) = 0$

3. RESULTS

3.1 Scenario setup

Before presenting each case, we begin with conceptual examples that illustrate how trust values evolve as nodes interact within the network.

3.1.1 Conceptual instance I

At the start of network operation (time t_0), all nodes are given an initial trust value of 0, while any node outside transmission ranges is assigned -∞, as shown in Eq. (6) and illustrated in Figure 2. The corresponding entries of the trust table of n_1 are listed in Table 3.

$$T(n_1, n_2) = T(n_1, n_3) = 0, \quad T(n_1, n_4) = -\infty \tag{6}$$

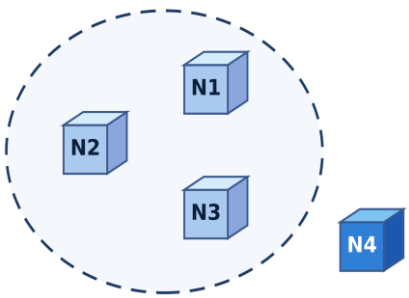


Figure 2. Initial trust assignment at time t_0

Table 3. Table entries of initial trust values of node n_1

Source Node	Destination Node	Trust Value
n_1	n_1	—
n_1	n_2	0
n_1	n_3	0
n_1	n_4	-∞

3.1.2 Conceptual instance II

When n_1 initiates communication with node n_2 at time t_1, as shown in Figure 3, it observes the five parameters of n_2 and obtains $x = 0.80$. Applying Eq. (5) to the neutral initial value gives the updated trust of Eq. (7), and the resulting table entries are listed in Table 4:

$$T(n_1, n_2) = 0 + 0.5(0.80 - 0) = 0.40 \tag{7}$$

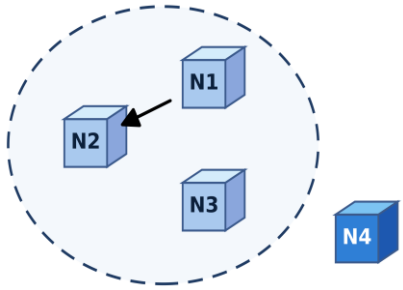


Figure 3. Connection initiated from n_1 to n_2 at time t_1

Table 4. Table entries of computed trust values of node n_1

Source Node	Destination Node	Trust Value
n_1	n_1	—
n_1	n_2	0.40
n_1	n_3	0
n_1	n_4	-∞

3.1.3 Conceptual instance III

When nodes n_2, n_3 and n_4 are within transmission range of n_1 at time t_2, as shown in Figure 4, the trust matrix is recomputed using Eq. (5). Node n_2 is observed a second time and its value moves half of the remaining distance towards x; node n_3 is observed for the first time; and node n_4 has entered range but has not yet been observed, so it is entered at the initial value and carries the unevaluated mark, as shown in Eq. (8) and listed in Table 5:

$$T(n_1, n_2) = 0.40 + 0.5(0.80 - 0.40) = 0.60, \quad T(n_1, n_3) = 0 + 0.5(0.80 - 0) = 0.40, \quad T(n_1, n_4) = 0 \tag{8}$$

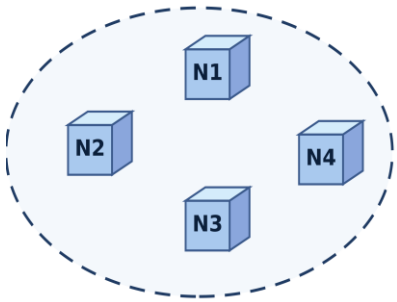


Figure 4. Nodes within transmission range of node n_1 at time t_2

Table 5. Table entries of computed trust values of node n_1

Source Node	Destination Node	Trust Value
n_1	n_1	—
n_1	n_2	0.60
n_1	n_3	0.40
n_1	n_4	0

3.1.4 Conceptual instance IV

This scenario considers cases in which multiple paths are available for selecting an intermediate node (see Figure 5). Case 1: Only one intermediate node has the highest trust value → It is directly selected. Case 2: When multiple nodes (e.g., n_2 and n_3) have converged to the same trust value (0.75), as shown in Table 6, the following selection step is based on the energy level.

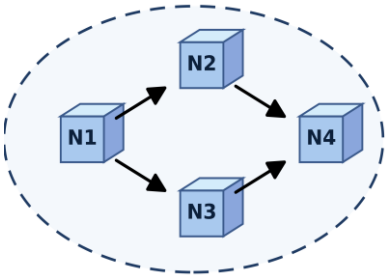


Figure 5. Two communication paths available from source node n_1 to destination node n_4

Table 6. Table entries of computed trust values of node n_1

Source Node	Destination Node	Trust Value
n_1	n_1	—
n_1	n_2	0.75
n_1	n_3	0.75
n_1	n_4	0

3.1.5 Conceptual instance V

If multiple nodes still have equal trust values, as compared in Table 7, the protocol chooses the node with the highest energy level. Energy levels represent the remaining battery life of the nodes as a percentage of their total capacity. For example, node n_2 has a trust value of 0.75 and an energy level of 40%, while node n_3 has a trust value of 0.75 and an energy level of 60%. Since n_3 has higher remaining energy, it is selected as the next hop. Complete entries appear in Table 8.

Table 7. Example energy-based tie-break comparison

Node	Trust Value	Energy Level
n_2	0.75	40%
n_3	0.75	60%

Table 8. Table entries of computed trust values of node n_1

Source Node	Destination Node	Trust Value	Energy Level
n_1	n_1	—	—
n_1	n_2	0.75	40%
n_1	n_3	0.75	60%
n_1	n_4	0	50%

3.2 Performance evaluation

The performance of TBAR-IoT was evaluated using OMNeT++ 6.0 with the INET framework. Fifty nodes were deployed randomly over a 1,000 m × 1,000 m area, each with a 450 m transmission range, moving according to the Random Waypoint model at speeds of up to 3 m/s. A simulation duration of 500 s was used to provide a sufficiently long observation period for route formation, trust updates and packet forwarding under attack. Ten percent of the nodes acted as blackhole attackers, dropping the packets they accepted. Traffic consisted of constant-bit-rate (CBR) flows between randomly selected source-destination pairs, with the offered load held fixed so that the comparison is not confounded by differences in demand. TBAR-IoT was compared with AODV, LDTS and Trust-AOMDV under identical conditions, and the same configuration was used for every result reported below; the only quantity varied is the proportion of malicious nodes, and only in the attack-intensity experiment reported at the end of this section. The evaluation metrics were PDR, energy consumption per delivered packet, and average end-to-end delay, none of which is derivable from the others.

As illustrated in Figure 6, at 10% malicious nodes, TBAR-IoT achieved the highest PDR at 91.2%, outperforming Trust-AOMDV (81.0%), LDTS (73.1%), and AODV (61.8%). This improvement stems from TBAR-IoT's adaptive multi-parameter trust computation, which leverages forwarding consistency, residual energy, and communication frequency to filter unreliable nodes during route formation, and from the procedure of Algorithm 3, which removes a confirmed attacker from the RC rather than merely lowering its trust value. The packet drop rate is not reported separately: under a fixed offered load it is the exact complement of the PDR and therefore conveys no information that Figure 6 does not already carry. Average end-to-end delay is reported in its place.

The energy consumption results in Figure 7 do not follow the same ordering as PDR. The ordering is consistent with the architectural differences between the protocols, although routing overhead was not measured separately in this study. TBAR-IoT recorded the lowest energy cost per delivered packet at 0.045 J. LDTS, despite trailing Trust-AOMDV in PDR, consumed less energy (0.049 J) because its hierarchical clustering aggregates trust locally within clusters and thereby limits network-wide control-message exchange. Trust-AOMDV, in contrast, recorded the highest energy cost (0.058 J): its multipath route discovery and maintenance process, which underpins its stronger PDR, maintains several candidate routes by design [20], and the discovery and maintenance traffic this entails is the most plausible explanation for its position relative to LDTS. AODV falls between the two (0.054 J) despite having no trust-computation overhead at all, a plausible reason being that a route traversing a blackhole node fails repeatedly and each failure triggers a fresh route discovery, so the energy spent is charged against a smaller number of delivered packets.

Figure 8 reports the average end-to-end delay. TBAR-IoT delivered packets in 90 ms on average, against 98 ms for Trust-AOMDV, 108 ms for LDTS and 121 ms for AODV. The ordering is again consistent with the architectural differences rather than measured directly. AODV incurs the highest delay, which is attributable to routes that traverse a blackhole node failing and having to be rediscovered, and every rediscovery adds a full route-request round trip to the packets queued behind it. LDTS pays a smaller but persistent cost for cluster formation and maintenance. Trust-AOMDV benefits from having alternative paths already available but spends time validating them before use. TBAR-IoT selects the next hop from trust parameters the node already holds, so no separate detection exchange lies on the forwarding path, a neighbor that fails the trust test is withheld from route selection immediately, and Algorithm 3 then completes the classification from observations already held and a single round of replies from the common neighbors, at the cost of a single rediscovery for the affected routes rather than a rediscovery per failed route request.

Furthermore, Figure 9 shows the effect of increasing malicious node ratios (0-20%) on PDR. TBAR-IoT declined only gradually, from 94.2% with no attackers to 85.2% at 20% malicious nodes, remaining above 85% throughout the tested range, whereas AODV degraded sharply from 88.8% to 36.8% over the same range. Trust-AOMDV and LDTS show intermediate degradation consistent with their respective

architectures, with Trust-AOMDV's multipath redundancy giving it more resilience than LDTS's cluster-dependent structure as the attacker ratio grows. This demonstrates the adaptability and robustness of TBAR-IoT under escalating attack intensity.

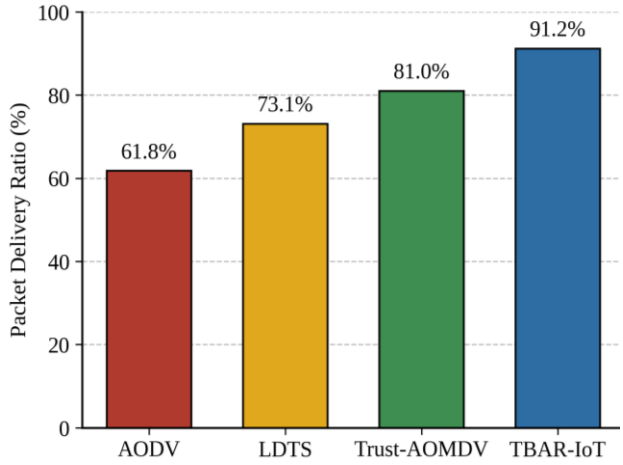


Figure 6. Packet delivery ratio (PDR) comparison across routing protocols

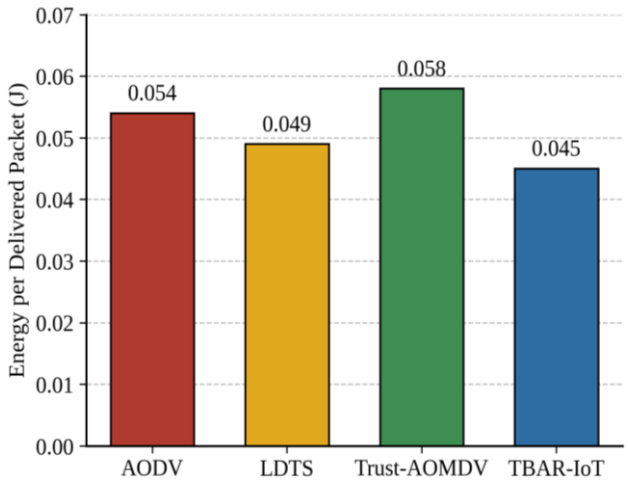


Figure 7. Energy consumption comparison across routing protocols

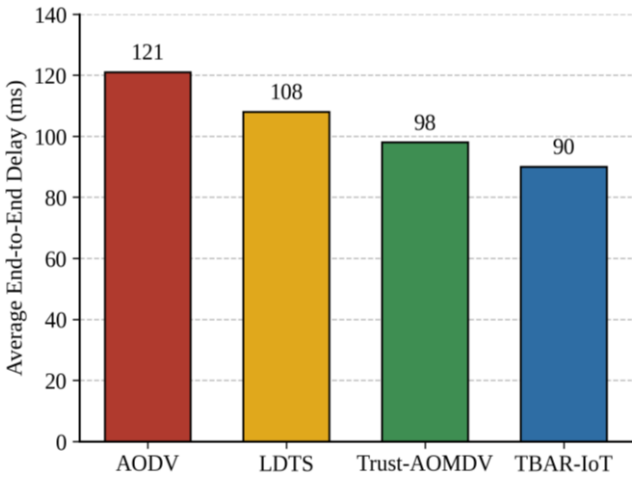


Figure 8. Average end-to-end delay comparison across routing protocols

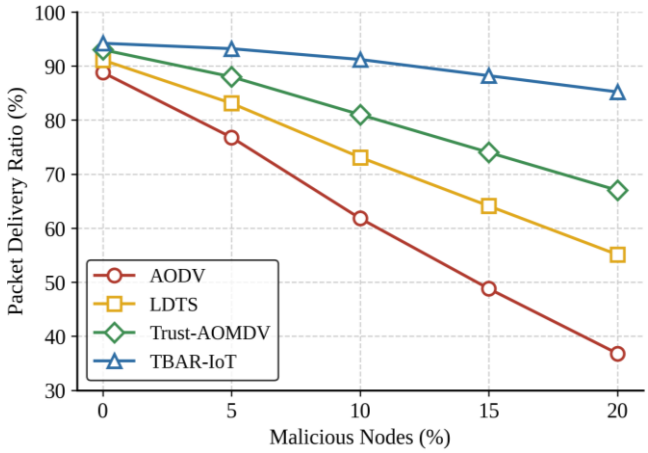


Figure 9. Effect of blackhole attack intensity on packet delivery ratio (PDR)

Overall, the results validate that TBAR-IoT offers a balanced combination of security, reliability, low latency, and energy efficiency compared to existing trust-based and non-trust-based routing protocols. TBAR-IoT employs a lightweight multi-parameter trust evaluation and provides energy-aware and resilient routing suitable for resource-constrained IoT environments.

4. CONCLUSIONS

This paper examined the performance of the proposed TBAR-IoT protocol and compared it with the existing routing methods AODV, LDTS, and Trust-AOMDV. The performance indicators used for the comparison were PDR, energy consumption per delivered packet, average end-to-end delay, and resilience to blackhole attacks.

The simulation results show that TBAR-IoT consistently achieves a higher PDR than both baseline and trust-based schemes. Under blackhole attack, the PDR of AODV decreases sharply because it has no trust awareness. TBAR-IoT, by contrast, maintains consistent packet delivery by isolating malicious nodes dynamically through behavioral trust assessment. Compared with LDTS, TBAR-IoT adapts better to varying network conditions, since LDTS relies on hierarchical trust structures that are sensitive to topology change. Trust-AOMDV improves delivery through multipath routing, but the additional routing overhead reduces its effectiveness in dense IoT settings. This same multipath overhead also explains why Trust-AOMDV, despite its relatively strong PDR, does not translate into energy savings; LDTS remains more energy-efficient than Trust-AOMDV precisely because its clustering limits control-message exchange, even though it delivers fewer packets successfully. TBAR-IoT reduces this trade-off in the evaluated scenario, combining the highest delivery ratio with the lowest energy cost per delivered packet among the protocols compared. The lowest average end-to-end delay was also recorded for TBAR-IoT, which is consistent with the reduced number of failed transmissions and route rediscoveries that follow from trust-aware next-hop selection. These observations support the conclusion that routing decisions based on adaptive behavioral trust can significantly improve network reliability without a corresponding increase in resource cost.

ACKNOWLEDGMENT

The authors wish to thank the Amity Institute of Information Technology, Amity University, Noida, India, for providing the research environment and computational resources required to carry out the simulations using OMNeT++.

REFERENCES

[1] Nassereddine, M., Khang, A. (2024). Applications of Internet of Things (IoT) in smart cities. In Advanced IoT Technologies and Applications in the Industry 4.0 Digital Economy, CRC Press, pp. 109-136. https://doi.org/10.1201/9781003434269-6

[2] Li, C., Wang, J., Wang, S., Zhang, Y. (2024). A review of IoT applications in healthcare. Neurocomputing, 565: 127017. https://doi.org/10.1016/j.neucom.2023.127017

[3] Zhang, L., Dabipi, I.K., Brown Jr, W.L. (2025). Internet of things applications for agriculture. In Internet of Things A to Z: Technologies and Applications, pp. 327-343. https://doi.org/10.1002/9781394280490.ch13

[4] Airehrour, D., Gutierrez, J., Ray, S.K. (2016). Secure routing for internet of things: A survey. Journal of Network and Computer Applications, 66: 198-213. https://doi.org/10.1016/j.jnca.2016.03.006

[5] Lim, K.S., Ooi, S.Y., Sayeed, M.S., Chew, Y.J., Ahmad, N.M. (2025). Securing the internet of things: Systematic insights into architectures, threats, and defenses. Electronics, 14(20): 3972. https://doi.org/10.3390/electronics14203972

[6] Hasan, A.A., Fang, X., Latif, S., Iqbal, A. (2025). Context-aware trust prediction for optimal routing in opportunistic IoT systems. Sensors, 25(12): 3672. https://doi.org/10.3390/s25123672

[7] Perkins, C., Belding-Royer, E., Das, S. (2003). Ad hoc on-demand distance vector (AODV) routing. RFC 3561, Internet Engineering Task Force. https://www.rfc-editor.org/info/rfc3561/.

[8] Johnson, D.B., Maltz, D.A. (1996). Dynamic source routing in ad hoc wireless networks. In Mobile Computing, Springer, pp. 153-181. https://doi.org/10.1007/978-0-585-29603-6_5

[9] Tseng, F.H., Chou, L.D., Chao, H.C. (2011). A survey of black hole attacks in wireless mobile ad hoc networks. Human-Centric Computing and Information Sciences, 1(1): 4. https://doi.org/10.1186/2192-1962-1-4

[10] Al-Kashoash, H. (2020). Congestion Control for 6LoWPAN Wireless Sensor Networks: Toward the Internet of Things. Springer, Cham. https://doi.org/10.1007/978-3-030-17732-4

[11] Khan, M.A., Rais, R.N.B., Khalid, O., Ahmad, S. (2024). Trust-based optimized reporting for detection and prevention of black hole attacks in low-power and lossy green IoT networks. Sensors, 24(6): 1775. https://doi.org/10.3390/s24061775

[12] Jøsang, A., Ismail, R., Boyd, C. (2007). A survey of trust and reputation systems for online service provision. Decision Support Systems, 43(2): 618-644. https://doi.org/10.1016/j.dss.2005.05.019

[13] Dhelim, S., Aung, N., Kechadi, M.T., Ning, H., Chen, L., Lakas, A. (2023). Trust2Vec: Large-scale IoT trust management system based on signed network embeddings. IEEE Internet of Things Journal, 10(1): 553-562. https://doi.org/10.1109/JIOT.2022.3201772

[14] Shoaei, Z., Esmaeilyfard, R., Javidan, R., Jalili, A. (2025). A new adaptive neuro-fuzzy trust-based routing for dynamic IoT networks. Cluster Computing, 28(11): 710. https://doi.org/10.1007/s10586-025-05381-2

[15] Li, X., Zhou, F., Du, J. (2013). LDTS: A lightweight and dependable trust system for clustered wireless sensor networks. IEEE Transactions on Information Forensics and Security, 8(6): 924-935. https://doi.org/10.1109/TIFS.2013.2240299

[16] Mohajerani, J., Mohammadi Ghanatghestani, M., Hashemipour, M. (2024). MCTE-RPL: A multi-context trust-based efficient RPL for IoT. Journal of Network and Computer Applications, 229: 103937. https://doi.org/10.1016/j.jnca.2024.103937

[17] Hassan, J., Sohail, A., Awad, A.I., Zaka, M.A. (2024). LETM-IoT: A lightweight and efficient trust mechanism for sybil attacks in internet of things networks. Ad Hoc Networks, 163: 103576. https://doi.org/10.1016/j.adhoc.2024.103576

[18] Dazine, J., Maizate, A., Hassouni, L. (2025). A new intrusion detection system for RPL based on machine learning. International Journal of Safety and Security Engineering, 15(6): 1089-1101. https://doi.org/10.18280/ijsse.150601

[19] Yousif, Y.K., Saeed, A.Q., Mohammed, O.H., Mostafa, S.A., Ali, R.R. (2026). A deep reinforcement learning-based trust management model for secure routing in vehicular ad-hoc networks under real-time traffic variations. International Journal of Safety and Security Engineering, 16(1): 17-27. https://doi.org/10.18280/ijsse.160102

[20] Govindaraj, M., Arumugam, S. (2023). Enhancing IoT security with trust-based mechanism for mitigating black hole attacks. International Journal of Safety and Security Engineering, 13(5): 917-923. https://doi.org/10.18280/ijsse.130515

[21] Sagar, S., Mahmood, A., Sheng, Q.Z., Zhang, W.E., Zhang, Y., Pabani, J.K. (2024). Understanding the trustworthiness management in the social internet of things: A survey. Computer Networks, 251: 110611. https://doi.org/10.1016/j.comnet.2024.110611

[22] Ahmadi, K., Javidan, R. (2024). A trust based anomaly detection scheme using a hybrid deep learning model for IoT routing attacks mitigation. IET Information Security, 2024(1): 4449798. https://doi.org/10.1049/2024/4449798

[23] Mohammad, A.A.K., Batta, K.B., Bari, M.A., et al. (2026). Trust-aware five-class multinomial logistic regression for packet-dropping mitigation in mobile ad hoc networks. International Journal of Safety and Security Engineering, 16(1): 1-15. https://doi.org/10.18280/ijsse.160101

[24] Sangeetha, K., Arulanandam, K. (2025). Adaptive trust-based secure routing protocol with reinforced anomaly detection for IoT networks. International Journal on Smart Sensing and Intelligent Systems, 18(1): 20250032. https://doi.org/10.2478/ijssis-2025-0032

[25] Rodríguez, A., López-Chau, A., Dávila-Nicanor, L., Landassuri-Moreno, V., Lazcano-Salas, S. (2026). Lightweight protection mechanisms for IoT networks based on trust modelling. IoT, 7(1): 18. https://doi.org/10.3390/iot7010018

[26] Shafique, A., Qureshi, I., Haseeb, K., Abbas, N., Khan,

A., Cheng, B. (2026). AI-based trust and decision modeling for scalable IoT networks with SDN optimization. Cluster Computing, 29(13): 743. https://doi.org/10.1007/s10586-026-06580-1

[27] Srinivasan, J. (2025). Innovative cross-layer defense mechanisms for blackhole and wormhole attacks in wireless ad-hoc networks. Scientific Reports, 15(1): 14747. https://doi.org/10.1038/s41598-025-97094-0

[28] Mutleg, M.L., Mahmood, A.M., Al-Nayar, M.M.J. (2024). A comprehensive review of cyber-attacks targeting IoT systems and their security measures. International Journal of Safety and Security Engineering, 14(4): 1073-1086. https://doi.org/10.18280/ijsse.140406

NOMENCLATURE

$T(n_i, n_j)$	Trust value held by node i in node j, $T \in [0, 1]$; $-\infty$ if n_j is out of range
w_i	Weight of each parameter
RC	Route Cache
x	Aggregate trust score of Eq. (2)
Z	Output of Eq. (3), $Z \in \{0, 1\}$
T_w	Observation window (T_w = 20 s)
T_h	Hold-down period ($T_h = 5T_w$)
A_{min}	Minimum forwarding samples, over consecutive windows, needed before classification (A_{min} = 10)
Trust Table	Per-neighbor record of the observations behind P_1-P_5
E	Residual energy of a node; P_4 is its normalized form
W	Neighbors of n_i that are also neighbors of n_j

Greek symbols

θ	Activation threshold (θ = 0.5)
η	Update rate of the trust value (η = 0.5)