International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2026/03 Volume 13, Issue 1, January – February (2026)

RESEARCH ARTICLE

EHO-CNN-TBR: A Trust-Based Secure Routing Model for IoT Using Elephant Herding Optimization and Convolutional Neural Networks

K. Vimala
Department of Computer Science, SRM Arts and Science College, Kattankulathur, Tamil Nadu, India.
✉ vimalajit12@gmail.com

M. Viju Prakash
School of Computing and Innovative Technologies, British University Vietnam, Vietnam.
viju.m@buv.edu.vn

A. Manikandan
Department of Computer Science and Engineering, Paavai Engineering College, Namakkal, Tamil Nadu, India.
manikandanarumugampec@paavai.edu.in

Prathap Mani
Department of Computer Science and Information Technology, American University of Kurdistan, Duhok, Kurdistan Region, Iraq.
prathap.mani@auk.edu.krd

Dinesh Mavaluru
Department of Information Technology, College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia.
d.mavaluru@seu.edu.sa

V. Porkodi
Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Sivas University of Science and Technology, Sivas, Turkey.
porkodiv1981@sivas.edu.tr

Received: 23 August 2025 / Revised: 18 November 2025 / Accepted: 28 December 2025 / Published: 26 February 2026

Abstract – The fast expansion of the Internet of Things (IoT) has enabled autonomous communication across massive, heterogeneous device networks. However, the dynamic topology and resource limitations of IoT environments expose routing processes to severe security threats, particularly malicious attacks that disrupt reliability and network stability. This study introduces EHO-CNN-TBR, a hybrid secure routing framework that integrates Elephant Herding Optimization (EHO) for adaptive trust evaluation with a Convolutional Neural Network (CNN) for real-time anomaly detection. EHO optimizes trust based on energy level, packet forwarding behavior, and historical performance, while CNN identifies malicious or abnormal traffic patterns. Using OMNeT++ with the INET Framework, the proposed model is evaluated against ACO-TBR [11] and FL-TBR [18]. Comprehensive results across PDR, delay, throughput, energy, routing overhead, and detection metrics demonstrate that EHO-CNN-TBR substantially outperforms existing approaches, offering a more reliable and attack-resilient routing solution for secure IoT communication.

Index Terms – Internet of Things, Secure Routing, Trust Management, Elephant Herding Optimization, Convolutional Neural Network, Routing Attacks.

1. INTRODUCTION

An extensive network of linked devices can now interact, gather, and share data on their own thanks to the Internet of Things (IoT), a game-changing technology. IoT is changing industries by offering real-time insights and operational efficiency, from smart homes and industrial automation to

RESEARCH ARTICLE

healthcare and environmental monitoring [1]. Due to developments in wireless communication, sensor downsizing, and cloud computing, it is anticipated that there will be tens of billions of IoT devices worldwide. The heterogeneous nature of devices and their resource limitations, such as limited processing power, battery life, and memory, provide major problems for IoT networks despite these developments [2]. Ensuring safe and dependable data packet routing across these networks is one of the most important concerns. IoT systems are particularly susceptible to a number of security risks, such as data manipulation, eavesdropping, and routing attacks like Sybil, wormhole, and blackhole attacks, as they frequently function in open and dynamic contexts.

These threats jeopardize data availability, confidentiality, and integrity in addition to impairing network performance [3]. In IoT networks, traditional routing algorithms frequently rely on static measurements and are not flexible enough to manage changing topologies and dynamic threats. These protocols are ill-equipped to fend off routing assaults because they usually assume a benign environment and lack tools to assess nodes' trustworthiness or identify unusual activity.

Routing attacks can seriously impair the effectiveness and security of data transmission. Examples of these attacks include wormhole attacks, which interfere with routing by establishing fake links, Sybil attacks, in which a single node assumes numerous identities, and blackhole attacks, in which malevolent nodes delete packets [4]. Packet loss, increased latency, and network splitting could result from hacked or malicious nodes continuing to participate in routing in the absence of strong trust measures.

Furthermore, current trust-based routing models frequently employ traditional trust computation techniques that are neither adaptive nor context-aware, or they rely on single-layer decision-making. Their ability to detect and mitigate threats in real time is hampered by these constraints. In order to develop a safe, scalable, and adaptable routing solution for IoT networks, a hybrid strategy that incorporates trust assessment, bio-inspired optimization, and intelligent anomaly detection is therefore desperately needed [5] [6].

Traditional routing protocols are growing less effective in the face of changing security risks as IoT networks continue to grow in size and complexity. These protocols frequently ignore nodes' dynamic behavior and the possible existence of malevolent actors, which leaves data transmission channels seriously vulnerable. The idea of trust-based routing has become more popular in such a decentralized and resource-constrained setting.

Trust metrics can offer important information about the dependability and long-term behavior of nodes. Nevertheless, a lot of the trust-based methods that are now in use rely on rule-based logic or static thresholds, which are not very flexible when it comes to evolving network conditions or clever attack techniques.

Intelligent, trust-aware, and adaptive routing systems that can both dynamically access node trustworthiness and instantly identify and react to malicious activity are becoming more and more necessary [7]. A strong, self-adaptive routing framework that guarantees network efficiency and security in IoT systems may be created by fusing deep learning-based anomaly detection with bio-inspired optimization algorithms.

1.1. Problem Statement

IoT networks face dynamic, heterogeneous, and resource-limited conditions that make traditional routing protocols ineffective against advanced attacks such as blackhole, Sybil, wormhole, and greyhole. Existing trust-based routing methods depend on static or rule-based trust computation, while machine learning–based anomaly detectors operate separately from routing decisions, creating gaps in real-time threat response.

Bio-inspired optimization algorithms like ACO and PSO improve efficiency but remain slow, vulnerable to manipulated metrics, and limited in detecting malicious nodes. These limitations lead to reduced packet delivery, increased delay, and persistent security vulnerabilities. Therefore, there is a pressing need for an adaptive, intelligent routing framework that integrates trust evaluation with real-time anomaly detection to ensure secure and efficient routing in IoT networks.

1.2. Objective

The goal of this research is to combine CNN and EHO to create a hybrid trust-based secure routing model for IoT networks. Based on variables such as energy level, packet forwarding behavior, and historical performance utilizing EHO, the model dynamically determines trust ratings for nodes. CNN, on the other hand, analyzes network traffic patterns to identify malicious activity and routing irregularities in real time. For safe and effective routing decisions, these two elements are combined into a single framework. Using simulation tools, the suggested EHO-CNN-TBR model is assessed and contrasted with other approaches, such as ACO-TBR [8] and FL-TBR [9].

This is how the remainder of the paper organized as structured: In Section 2, relevant research on intelligent and trust-based routing techniques for Internet of Things networks is reviewed. The suggested EHO-CNN-TBR approach is presented in Section 3. This includes the system architecture, the routing decision mechanism, the Convolutional Neural Network (CNN) for anomaly detection, and the usage of Elephant Herding Optimization (EHO) for trust computation. Section 4 offers a thorough performance review and describes

RESEARCH ARTICLE

the experimental setting. Section 5 wraps up the report and provides guidance for further research.

2. RELATED WORKS

Securing routing in Internet of Things (IoT) networks has become increasingly challenging due to dynamic topology, heterogeneous devices, and sophisticated cyberattacks. Existing literature highlights multiple approaches—including trust computation, bio-inspired optimization, and machine-learning-based anomaly detection—to enhance routing reliability. However, many studies either lack real-time adaptability, fail to detect coordinated attacks, or introduce high computational overhead.

2.1. Trust-Based Routing Models

Trust-based routing has emerged as a vital strategy to counter malicious nodes and unreliable communication. Bao and Chen (2021) introduced a dual-component trust framework evaluating communication and data trust. Their model improved resistance against basic misbehavior but failed to handle sophisticated coordinated attacks due to static trust thresholds [8].

Chen et al. (2021) later proposed an adaptive trust mechanism incorporating dynamic trust updates and behavioral metrics. While this improved responsiveness, the model suffered from high computational overhead in dense networks [9]. Rahman et al. (2022) developed a multi-metric trust-based routing protocol that integrates packet drop ratio, energy, and historical reliability. Their method enhanced packet delivery and reduced misrouting events; however, the absence of anomaly detection made it ineffective against Sybil and collaborative blackhole attacks [10].

2.2. Bio-Inspired Optimization for IoT Routing

Bio-inspired algorithms have been widely applied to routing optimization due to their decentralized nature and adaptability. Ant Colony Optimization-based Trust Routing (ACO-TBR) [11][12] dynamically updates trust pheromones according to node interactions. Although effective in small networks, ACO suffers from slow convergence, susceptibility to pheromone manipulation, and increased routing overhead under attack scenarios.

Elephant Herding Optimization (EHO) has gained traction for its fast convergence and clan-based adaptive search mechanism. Kumar et al. (2023) demonstrated EHO's ability to optimize energy-efficient clustering, significantly improving network lifetime. However, its application to IoT security remains limited, and existing EHO implementations do not consider anomaly behaviors or adversarial patterns affecting trust. Hybrid PSO–Fuzzy or PSO–ACO approaches improved path selection but still failed to detect malicious nodes autonomously [13] These models operate solely on performance metrics without incorporating behavioral intelligence, making them vulnerable when adversaries mimic normal node behavior.

2.3. Machine Learning and Deep Learning-Based Anomaly Detection

Machine learning (ML), especially deep learning (DL), plays a crucial role in securing IoT systems by detecting nonlinear malicious patterns that traditional trust models cannot identify. Li et al. (2022) applied a CNN-based IDS to IoT traffic, showing significant gains over classical ML models. However, CNN inference overhead makes standalone CNN unsuitable for real-time routing [14]. Diro and Chilamkurti (2021) utilized CNNs for distributed intrusion detection, achieving low false positives but requiring heavy computation [15]. Singh et al. (2023) introduced a CNN–LSTM hybrid model for botnet detection, improving recall but demanding large training data and high resources [16]. Similarly, Al-Tamimi et al. (2024) proposed a lightweight 1D-CNN for anomaly detection, achieving faster inference but lacking integration with routing protocols. Despite promising results, these studies treat anomaly detection and routing as separate tasks, limiting their ability to prevent malicious nodes from participating in actual routing operations [17].

2.4. Hybrid Trust–Optimization–ML Routing Models

Recent research emphasizes integrating trust evaluation with optimization algorithms and machine learning intelligence. Fuzzy Logic-based Trust Routing (FL-TBR) [18] provides flexible trust representation but suffers from rule explosion and subjective tuning. Hybrid PSO–Fuzzy trust models (2024) enhanced trust accuracy but remained prone to misclassification in dynamic attack environments. A significant research gap persists: no existing model combines trust optimization (via metaheuristics) with deep-learning anomaly detection to simultaneously assess trust, detect malicious behavior, and drive routing decisions [18].

Existing literature fails to deliver a unified, adaptive, and scalable approach capable of maintaining both security and performance under high attack intensity. This gap motivates the development of EHO-CNN-TBR, which fuses EHO-based trust optimization with CNN-driven anomaly detection, delivering real-time malicious node isolation and trust-aware routing decisions [19]. Hybrid PSO–Fuzzy trust routing models combine Particle Swarm Optimization (PSO) with fuzzy logic to improve trust estimation in IoT networks. PSO optimizes trust parameters by exploring multiple routing solutions, while fuzzy logic manages uncertainty in node behavior. Although these models enhance trust accuracy, they suffer from misclassification under dynamic attack conditions and increased computational overhead, which limits scalability in dense IoT environments [20].

Zhang et al. (2023) proposed a swarm-intelligent routing approach that utilizes collective behavior-based optimization

RESEARCH ARTICLE

for improved path exploration in IoT networks. While the integrated trust and security mechanisms, making it vulnerable to routing attacks such as blackhole and Sybil attacks [21]. A summary of related trust-based and optimization-driven routing models is presented in Table 1. Table 1 shows existing trust-based routing models along with model improves adaptability and route diversity, it lacks their algorithms, advantages, and disadvantages. These approaches demonstrate the strengths of trust-based routing but fall short in scalability and adaptability under modern IoT attack vectors. This gap motivates the development of hybrid solutions like the proposed EHO-CNN-TBR.

Table 1 Summary of Related Trust-Based and Optimization-Driven Routing Models

Author & Year	Method / Model	Algorithm Used	Advantages	Disadvantages
Bao & Chen (2021) [8]	Dual Trust Model	Behaviour-based trust	Simple and effective for basic threats	Static trust; ineffective for advanced attacks
Chen et al. (2021) [9]	Adaptive Trust Evaluation	Dynamic trust algorithm	Fast adaptation to node behaviour	High computational cost
Rahman et al. (2022) [10]	Multi-Metric Trust Routing	Weighted trust	Good PDR and reliability	No anomaly detection
ACO-TBR (2023) [11]	Trust-Based Routing	Ant Colony Optimization	Adaptive route discovery	High convergence time
ACO-TBR (2023) [12]	Trust-Based Routing	Ant Colony Optimization	Improved trust propagation	Pheromone manipulation risk
Kumar et al. (2023) [13]	Energy-Efficient Routing	Elephant Herding Optimization (EHO)	Fast convergence; low energy consumption	No security consideration
Mohan et al. (2024) [14]	PSO-Fuzzy Trust Routing	PSO + Fuzzy Logic	Improved trust accuracy	Poor scalability
Li et al. (2022) [15]	IoT Intrusion Detection	CNN	High detection accuracy	High computational overhead
Diro & Chilamkurti (2018) [16]	Deep Learning IDS	CNN	Low false positives	Heavy resource usage
Singh et al. (2023) [17]	Botnet Detection	CNN-LSTM	Strong recall	High training cost
Al-Tamimi et al. (2024) [18]	Smart Home IDS	1D-CNN	Lightweight and fast	No routing integration
FL-TBR (2024) [19]	Trust Routing	Fuzzy Logic	Good uncertainty handling	Subjective rule design
Hybrid PSO–Fuzzy (2024) [20]	Hybrid Trust Routing	PSO + Fuzzy	Better trust estimation	Misclassification under dynamic attacks
Zhang et al. (2023) [21]	Swarm-Intelligent Routing	Swarm intelligence algorithms	Better path exploration	Inadequate security mechanisms

3. PROPOSED METHODOLOGY: EHO-CNN-TBR

3.1. System Architecture

For safe and effective communication in Internet of Things networks, the suggested EHO-CNN-TBR model combines Elephant Herding Optimization (EHO) and Convolutional Neural Networks (CNN) into a hybrid trust-based routing framework as shown in Figure 1. In order to enable dynamic, attack-resistant routing decisions, the architecture is built to manage both anomaly detection and trust evaluation. There are multiple essential elements that make up the suggested hybrid trust-based secure routing model. IoT nodes are diverse devices placed throughout the network that carry out functions like data relaying, transmission, and sensing. Every

RESEARCH ARTICLE

node keeps track of local information such as its energy level, packet forwarding history, and interactions with nearby nodes. Each neighbor's trust score is determined by the Trust Evaluation Module (EHO Engine) using factors such residual energy, packet forwarding success rate, and past behavior (such as packet drops or delays). By simulating the behavior of elephant clans, in which nodes follow matriarchs that approximate ideal routing behavior, the Elephant Herding Optimization (EHO) algorithm improves these trust levels. In routing, nodes with higher trust scores are given priority.

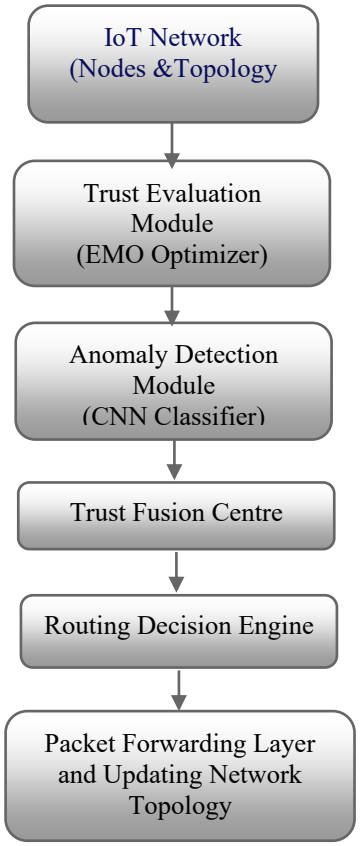

Figure 1 Proposed EHO-CNN-TBR Architecture

The Anomaly Detection Module (CNN Engine) categorizes nodes as malicious or benign based on real-time traffic patterns. It makes use of a Convolutional Neural Network (CNN) that has been trained to analyze characteristics such as erratic packet rates, abrupt route changes, and indications of previous misconduct. The output, which informs and modifies the node's trust score, is either a binary classification or an anomaly score. By integrating information from the CNN-based anomaly detection and the EHO trust evaluation, Trust Fusion Center acts as an integrative unit. It eliminates malicious nodes and modifies the final trust score. Using trust-weighted metrics, the Routing Decision Engine chooses the most secure and effective routing path, guaranteeing that only trustworthy, non-malicious nodes take part in data forwarding. The simulation environment for evaluating the model is provided by the Communication Layer (INET/OMNeT++). To verify the suggested method, it logs performance metrics, controls packet flows, and modifies network architecture.

3.2. Elephant Herding Optimization (EHO) for Trust Computation

Based on the social behavior of elephants in the wild, the Elephant Herding Optimization (EHO) algorithm is a metaheuristic inspired by nature. In this study, the trust levels of IoT nodes are calculated and optimized for safe and intelligent routing. Clan splitting and updating are two important behavioral features that the EHO algorithm mimics [21]. Each individual elephant serves as a node in the context of IoT routing, and an elephant's fitness is a measure of that node's reliability. The overall objective is to move the population (IoT nodes) in the direction of routes that are resilient to attacks, energy-efficient, and high-trust.

Step 1: Initialization and Grouping of Nodes

Suppose the IoT network has N nodes. Each node is treated as a candidate solution (i.e., an elephant), and Each node is defined by a position vector representing its parameters, as defined in equation (1). The grouping of nodes into clans is defined in equation (2).

$$X_i = [E_i, F_i, H_i] \quad (1)$$

Where, E_i = Residual energy of node i, Fi = Packet forwarding ratio of node i, Hi = Historical behavior score of node i. These nodes are divided into K clans, where each clan has a number of elephants (nodes), forming subgroups of the network.

$$Clan_k = \{X_{k,1}, X_{k,2}, \ldots\ldots, X_{k,m}\}, k = 1,2, \ldots, K \quad (2)$$

Step 2: Clan Updating Operator

Within each clan, the matriarch elephant (i.e., the node with the highest trust) guides the movement (update) of other elephants. For each non-matriarch elephant in clank, the position update of non-matriarch nodes is performed using the clan updating operator defined in equation (3).

$$X_{k,i}^{(t+1)} = X_{best,k}^{(t)} + a.\left(r.\left[X_{best,k}^{(t)} - X_{k,i}^{(t)}\right]\right) \quad (3)$$

Where, $X_{k,i}^{(t+1)}$ is the current position (trust feature $X_i = [E_i, F_i, H_i]$ vector) of the i^{th} elephant in the k^{th} clan at iteration t,$X_{best,k}^{(t)}$ is the position of the clan matriarch (highest fitness),α is a scaling factor (typically 0.5) and r is a random number in [0, 1] to introduce stochasticity. This operator helps nodes move toward more trustworthy configurations, improving overall trust reliability.

Step 3: Separating Operator

RESEARCH ARTICLE

To avoid premature convergence and maintain diversity, some elephants (nodes) are separated from their clans and reinitialized randomly in the search space. Let the last elephant in each clan be the separated elephant, the separating operator used to maintain population diversity is defined in equation (4):

$$X_{worst,k}^{(t+1)} = X_{min} + r.(X_{max} - X_{min}) \quad (4)$$

Where,$(X_{max} - X_{min})$ are the lower and upper bounds of node trust parameters and r is a uniform random number between 0 and 1. This step introduces exploration and avoids local optima by allowing new node configurations to emerge.

Step 4: Fitness Function Design for Trust Evaluation

The fitness function evaluates the trustworthiness of each node based on a combination of features:

- Residual Energy $[E_i]$: Nodes with higher remaining energy are preferred.
- Packet Forwarding Ratio $[F_i]$: Ratio of successfully forwarded packets to received packets.
- Historical Behavior $[H_i]$: Record of past misbehaviors or anomalies.

The normalized weighted trust fitness function is defined in equation (5).

$$Fitness(X_i) = \omega_1.\frac{E_i}{E_{max}} + \omega_2.F_i + \omega_3.H_i \quad (5)$$

Where, $\omega_1, \omega_2, \omega_3$ are weights satisfying $\omega_1 + \omega_2 + \omega_3 = 1$, Each component is normalized to [0,1] range and Higher fitness implies higher trustworthiness. Typical weight values might be $\omega_1 = 0.3$: Energy, $\omega_2 = 0.5$: Forwarding behavior and $\omega_3 = 0.2$: Historical reputation

The fitness scores are used to Guide clan updating (matriarch selection), Update trust scores and influence the routing decision engine in selecting next-hop nodes

3.2.1. Flow of EHO in Trust Computation

The node population is initialized and divided into clans as the first step in the Elephant Herding Optimization (EHO) process for trust computation. A predetermined trust function is used to assess each node's initial fitness based on variables including energy level and past behavior.

The clan updating operator, which mimics the impact of the matriarch (leader) node in each clan, is then used to update the locations of non-matriarch nodes. To improve variety and prevent local optima, the separation operator is then applied to each clan's lowest-performing node. Following these modifications, all nodes' fitness values are recalculated. Until the algorithm converges or a predetermined number of iterations is reached, this process iterates. Secure and dependable paths in the IoT network are then chosen using the final trust scores obtained from this optimization [22][23][24].

3.3. CNN for Anomaly Detection

The EHO-CNN-TBR model's Convolutional Neural Network (CNN) component is used to detect anomalies in IoT networks in real time. In order to detect hostile or compromised nodes that might try routing assaults (such as sinkhole, greyhole, or blackhole attacks), this module learns from network traffic patterns and node activity.

Step 1: Input Features and Data Preprocessing

To train the CNN effectively, relevant features must be extracted and normalized. These features are gathered from node behavior and network traffic statistics.

Input Features: The input feature vector for node i at time t is defined in equation (6).

$$X_i^{(t)} = [x_1, x_2, x_3, \dots, x_n] \quad (6)$$

Where $x_j$represents the normalized value of thej^{th} feature.

Preprocessing Steps:

- Normalization: All feature values are scaled to a range [0,1] feature normalization is performed using the Min–Max method defined in equation (7).

$$x_j' = \frac{x_j - min(x_j)}{max(x_j) - min(x_j)} \quad (7)$$

- Reshaping: The feature vector is reshaped into a 2D format (matrix) if necessary to fit the CNN input shape (e.g., 1×n or n×1).
- Labeling: Data samples are labeled as:0 → Benign node and 1→ Malicious node

Step 2: CNN Architecture

A simple 1D CNN is well-suited for processing time-series or vectorized features in IoT anomaly detection. The following architecture is typically adopted.

Table 2 1D CNN Architecture for IoT Anomaly Detection

Layer	Configuration
Input Layer	Size = number of features (e.g., 7 features)
Conv1D Layer	Filters = 32, Kernel size = 3, Activation = ReLU
MaxPooling1D Layer	Pool size = 2
Conv1D Layer	Filters = 64, Kernel size = 3, Activation = ReLU

RESEARCH ARTICLE

MaxPooling1D Layer	Pool size = 2
Flatten Layer	Converts 2D matrix into 1D vector
Dense Layer	Units = 64, Activation = ReLU
Dropout Layer	Rate = 0.5 (for regularization)
Output Layer	Units = 1, Activation = Sigmoid (binary classification)

As shown in Table 2, the proposed 1D CNN architecture consists of two convolutional layers followed by max-pooling, a fully connected dense layer, and an output layer. The Rectified Linear Unit (ReLU) activation function is employed in all convolutional and dense layers to introduce non-linearity and enhance learning capability. The sigmoid activation function is used in the output layer to perform binary classification, where 0 indicates benign traffic and 1 indicates malicious traffic.

Activation Functions:

The ReLU (Rectified Linear Unit) activation function is defined in equation (8).

$$R_eLU(x) = \max(0, x) \quad (8)$$

Used in hidden layers for non-linearity.

The sigmoid activation function used for binary classification is defined in equation (9).

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (9)$$

Used in the output layer for binary classification (0 = benign, 1 = malicious).

Step 3: Training and Classification Methodology

Training Process:

Dataset: Use labeled data collected from simulation or real IoT scenarios containing both benign and malicious node behaviors.

Loss Function: The binary cross-entropy loss function used for CNN training is defined in equation (10).

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i \log(\hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i) \quad (10)$$

Where,y_i is the true label and $\hat{y}_i$ is the predicted probability

Optimizer: Use the Adam optimizer for fast convergence.

Batching and Epochs: Batch size: 32 or 64Epochs: Typically, 50–100 depending on convergence after training, for each node, the CNN predicts as shown in equation (11).

$$\hat{y}_i = \sigma(z) \quad (11)$$

If $\hat{y}_i > 0.5$,the node is classified as malicious, else benign.

These predictions are fed back into the Trust Fusion Center, where they influence the final trust value of nodes and determine whether a node is included in routing decisions [25][26].

3.4. Routing Decision Process

The suggested EHO-CNN-TBR model's routing decision is made to take advantage of the anomaly detection outputs produced by the Convolutional Neural Network (CNN) as well as the trust scores optimized by the Elephant Herding Optimization (EHO) algorithm.

This combination guarantees that routing decisions are resistant to malicious assaults and trust-aware.

Step 1: Combining Outputs from EHO and CNN

Each node i in the network has two key metrics after processing:

The optimized trust score obtained from EHO is defined in equation (12).

$$T_i \in [0,1] \quad (12)$$

Represents the optimized trustworthiness of node i, where 1 means fully trusted and 0 means untrusted. The anomaly classification output from CNN is defined in equation (13).

$$A_i = \begin{cases} 1, \text{if node i is classified as malicious} \\ 0, \text{if node i is classified as benign} \end{cases} \quad (13)$$

The combined Effective Trust Score (ETS) for node i integrates both the trust value and anomaly detection result as defined in equation (14).

$$ETS_i = T_i \times (1 - A_i) \quad (14)$$

If the CNN flags node i as malicious$(A_i - 1)$, then$ETS_i = 0$, effectively removing it from routing consideration. If the node is benign$(A_i - 0)$, then$ETS_i = T_i$, allowing the trust score to determine routing preference.

Step 2: Trust-Aware Routing Path Selection

Consider a path P in the IoT network represented as an ordered set of nodes, defined in equation (15).

$$P = \{n_1, n_2, \ldots, n_m\} \quad (15)$$

The trustworthiness of the entire path P is computed as the product of effective trust scores of all nodes along the path as defined in equation (16).

$$T_P = \prod_{j=1}^{m} ETS_{nj} \quad (16)$$

This multiplicative model ensures that a low-trust or malicious node drastically reduces the overall path trust.

Alternatively, a logarithmic sum can be used to avoid numerical underflow as defined in equation (17).

RESEARCH ARTICLE

$$\log\left(T_P = \sum_{j=1}^{m} \log(ETS_{nj})\right) \quad (17)$$

Step 3: Route Selection Criteria

From all possible routing paths $\{P_1, P_2, \ldots, P_k\}$, the routing engine selects the path P∗ that maximizes the combined trust metric while considering other network performance factors such as latency and energy consumption as defined in equation (18).

$$P^* = \arg\max_P(\beta . T_P - \gamma . D_P - \delta . E_P) \quad (18)$$

Where, T_P = Trustworthiness of path P, D_P = Estimated end-to-end delay of path P, E_P = Estimated energy cost for path &,β,γ, δ are weighting factors to balance trust, delay, and energy efficiency (with β+γ+δ=1).

Step 4: Routing Table Update

- The routing decision engine updates routing tables dynamically based on the above trust-aware metric.
- Nodes with ETS_i below a threshold τ (e.g., 0.5) are excluded from routing paths.
- Routes passing through suspected malicious nodes with$ETS_i = 0$ are discarded.
- Trust scores and anomaly outputs are periodically updated and routes recalculated to adapt to network changes and attacks.

This routing decision process results in an adaptive, secure, and performance-aware routing strategy that dynamically excludes malicious nodes and prioritizes trustworthy and efficient paths for IoT communication.

The complete working procedure of the proposed EHO-CNN-TBR model is summarized in Algorithm 1.

Input:

N = Set of IoT nodes

P = Set of network paths

FeatureSet = Traffic and trust parameters from nodes

CNN_model = Pretrained CNN for anomaly detection

max_iterations = Number of iterations for EHO

Output:

Secure routing path P*

Begin

1. // Step 1: Feature Extraction

For each node i ∈ N:

Extract features F_i = [energy, packet_forward_ratio, delay, drop_rate, etc.]

2. // Step 2: Anomaly Detection Using CNN

For each node i ∈ N:

A_i = CNN_model.predict(F_i)

If A_i == 1:

Mark node i as malicious

Else:

Mark node i as benign

3. // Step 3: Trust Computation Using EHO

Initialize elephant population (nodes) and group into clans

For iter = 1 to max_iterations do:

For each clan C:

Compute clan centre

For each elephant e in C:

Update position using clan updating operator

Apply separating operator for worst-performing elephant

Compute trust score T_e using fitness function:

T_e = w1 * PacketForwardRatio + w2 * (1 - DropRate) + w3 * (1 - Delay)

End For

// Step 4: Effective Trust Score Calculation

For each node i ∈ N:

If A_i == 1:

ETS_i = 0

Else:

ETS_i = T_i

// Step 5: Routing Decision

For each candidate path Pj∈ P:

Compute path trust TP_j = ∏ ETS_i for i ∈P_j

Compute delay DP_j and energy cost EP_j

Select optimal path P*:

P* = argmax (β * TP_j- γ * DP_j - δ * EP_j)

// Step 6: Data Transmission

Transmit data over path P*

// Step 7: Periodic Updates

RESEARCH ARTICLE

Periodically repeat steps 1–6 to adapt to changing network conditions

End

Algorithm 1 EHO-CNN-TBR Framework

Variables and Notation,

- A_i: CNN classification output (0 = benign, 1 = malicious).
- T_i: Trust score from EHO.
- ETS_i: Effective Trust Score.
- TP_j: Trust score of path P_j.
- β, γ, δ: Weighting coefficients for trust, delay, and energy.
- w1, w2, w3: Weights in EHO fitness function.

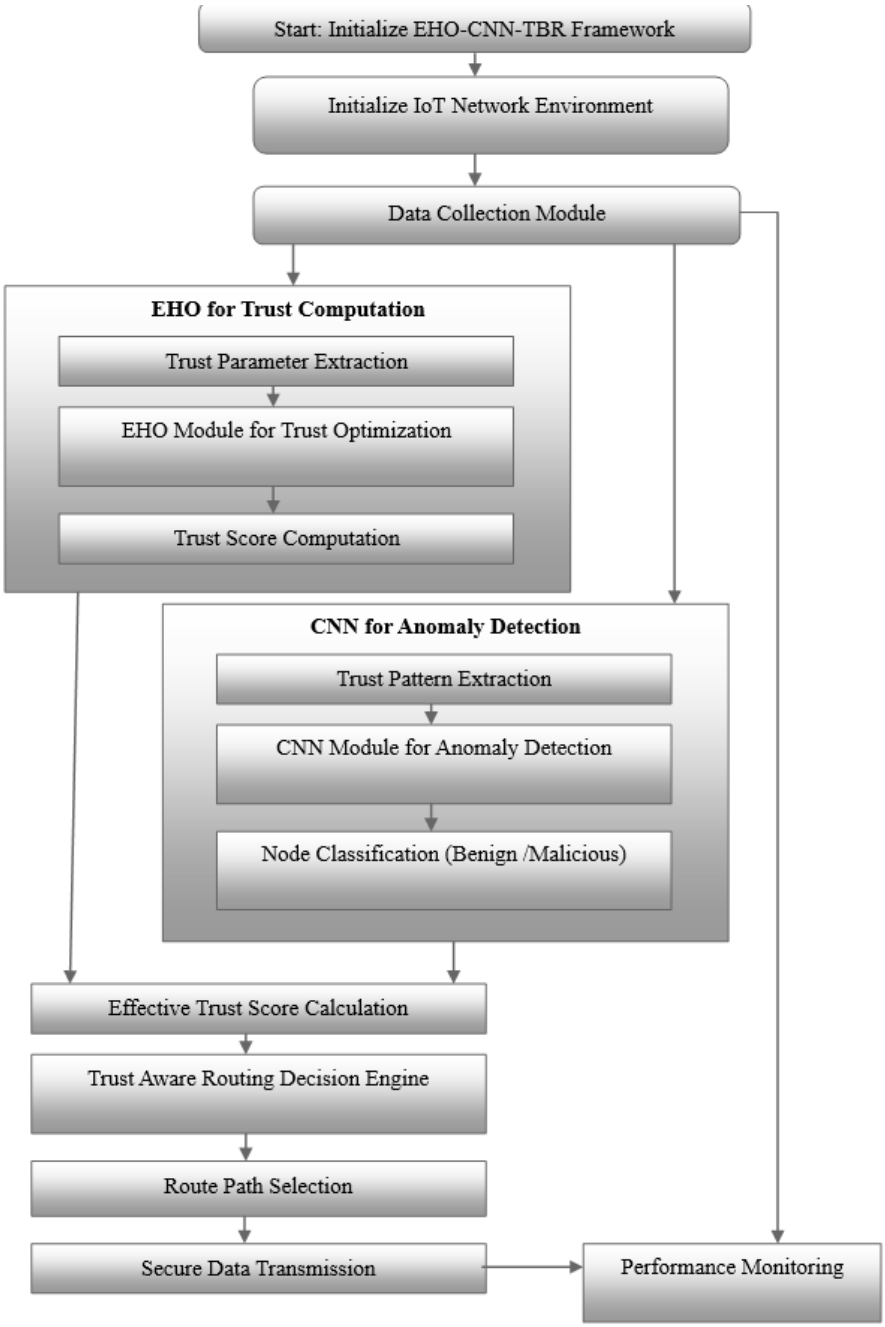

Figure 2 EHO-CNN-TBR Framework

RESEARCH ARTICLE

Figure 2 shown, The EHO-CNN-TBR Framework is a hybrid intelligent model designed to secure routing in Internet of Things (IoT) networks by dynamically identifying trustworthy nodes and filtering out malicious ones. It combines the strengths of bio-inspired optimization and deep learning to create a robust routing mechanism that adapts to the dynamic and often vulnerable nature of IoT environments. Energy levels, packet forwarding behavior, packet drop rates, and other network-related metrics are among the pertinent attributes that are extracted from each IoT node by the framework's first data collecting module. The CNN classifier and the EHO engine are the two main modules that receive these features concurrently. The EHO algorithm groups nodes into "clans" by imitating the social behavior of elephants in herds. It then iteratively updates the nodes' trust scores using a fitness function that assesses their dependability based on past and present behavior.

The EHO model dynamically modifies trust scores by rewarding consistently trustworthy nodes and penalizing unreliable ones through clan update and operator separation. The CNN module analyzes traffic patterns concurrently in order to identify irregularities. It uses supervised learning from past attack data to categorize each node as either benign or malicious. The CNN's output is binary; if a node is determined to be malicious, its impact on routing choices is removed. Each node's Effective Trust Score (ETS) is then calculated using the CNN's anomaly labels and the EHO's trust scores. A node's ETS is set to zero if the CNN flags it as malicious; otherwise, it equals the optimized trust score from EHO. The routing decision engine makes use of these ETS values when assessing all potential network paths and choosing the most trustworthy one overall, taking into account variables like end-to-end latency and energy efficiency. Thus, a weighted function that balances energy cost, latency, and trust is used to choose the final routing path. Data is then transmitted via this attack-resistant, trust-aware channel. The EHO-CNN-TBR framework incorporates a feedback loop to guarantee flexibility, updating trust assessments and anomaly classifications on a regular basis in response to fresh data. This makes it possible for the framework to react to node behavior changes and new security concerns in an efficient manner. All things considered, EHO-CNN-TBR provides a complete, clever solution for safe and effective routing in Internet of Things networks, guaranteeing high packet delivery, minimal latency, and resistance to routing attacks.

4. RESULT AND DISCUSSION

4.1. Experimental Setup

OMNeT++ version 6.0 integrated with the INET Framework, which is specifically made for modeling wireless networks like IoT environments, is used to simulate the suggested EHO-CNN-TBR (Elephant Herding Optimization with Convolutional Neural Network for Trust-Based Routing) framework. A Windows 10 computer with an Intel Core i5 CPU (2.2 GHz), 8 GB of RAM, and enough disk space to support extensive simulations and training data is used to conduct the tests. The University of New South Wales (UNSW) provided the Bot-IoT dataset, which is used to model real-world IoT traffic patterns, including both benign and malevolent assaults like DoS, DDoS, and data exfiltration. With more than 70 million labeled records, this dataset provides a wealth of features, including packet sizes, flow durations, and flags, which are essential for the CNN module's anomaly detection and trust assessment. The EHO-CNN-TBR model's performance is assessed by comparing it to two baseline algorithms: FL-TBR (Fuzzy Logic-Based Trust Routing) and ACO-TBR (Ant Colony Optimization-Based Trust Routing). Each node is initially given a trust rating of 0.5, and the comparison is carried out under different network densities: 75, 150, 225, and 300 total nodes. All nodes function in a dynamic environment across 100 simulation rounds in the 1000 m x 1000 m simulated IoT environment to evaluate adaptation in changing circumstances.

Table 3 General Simulation Parameters

Parameter	Value
Number of Nodes	75, 150, 225, 300
Network Area	1000 m × 1000 m
Transmission Range	250 m
Initial Energy	120 Joules
Packet Size	2 MB
MAC Protocol	IEEE 802.11
Antenna Type	Omni-Directional
Propagation Model	Two-Ray Ground
Number of Rounds	100
Initial Trust Value	0.5 per node

Table 3 summarizes the general simulation settings used in OMNeT++ for all routing schemes. The algorithm 1 parameters of the proposed EHO-CNN-TBR model and the comparative ACO-TBR and FL-TBR approaches are detailed in Tables 4 and 5, respectively. The algorithm 1 parameters for the proposed EHO-CNN-TBR framework and the comparative models are carefully chosen to ensure optimal performance and fair benchmarking across different routing approaches in IoT networks.

The algorithm for the EHO-TBR (Elephant Herding Optimization for Trust-Based Routing) module is set up with five clans, each of which has ten elephants. To balance

RESEARCH ARTICLE

exploration and exploitation, the separating operator rate is kept at 0.2 and the clan updating factor (α) is set to 0.5. The trust fitness function is assessed using a weighted combination of factors, especially $w_1 = 0.4$ for energy efficiency, $w_2 = 0.3$ for packet forwarding ratio, and $w_3 = 0.3$ for historical trust behavior. The optimization procedure is carried out for a maximum of 100 iterations. Twenty carefully chosen traffic features are supplied into the CNN (Convolutional Neural Network) module, which is used for anomaly identification. Several layers make up the architecture, which goes as follows: Input → Convolution → ReLU → Max Pooling → Fully Connected → Softmax. Two convolutional layers with a 3x3 kernel size make up the CNN, which is followed by two 2x2 max pooling layers. The Adam optimizer is used to optimize the model, and the ReLU activation function is used across the network. Training is carried out with a batch size of 128 and a learning rate of 0.001 across 30 epochs. Each simulation uses 50 ants for the comparative Ant Colony Optimization-Based Trust Routing (ACO-TBR) algorithm. With α (trust importance) and β (path quality importance) set at 1.0 and 2.0, respectively, the pheromone decay rate (ρ) is set at 0.3. The algorithm runs for a maximum of 100 times.

Finally, the three primary input variables used by the FL-TBR (Fuzzy Logic-Based Trust Routing) model are packet forwarding ratio (PFR), end-to-end delay, and energy consumption. Triangular membership functions classified as Low, Medium, and High are used to process them. The Mamdani method is used to do the fuzzy inference, and the centroid defuzzification strategy is used to determine the final trust value. Nine fuzzy rules make up the rule foundation that governs the system and determines the trustworthiness of nodes. When combined, these parameter setups provide a thorough and reliable assessment of safe, trust-based routing techniques designed for resource-constrained and dynamic IoT contexts.

4.2. Bot-IoT Dataset

A thorough resource for assessing network behavior in IoT environments is the Bot-IoT dataset, which was created by the University of New South Wales (UNSW) in partnership with the Australian Centre for Cyber Security. It includes a wide range of traffic situations, including both routine business activities and several kinds of cyberattacks, including data theft, distributed denial of service (DDoS), denial of service (DoS), and information collecting.

This dataset is especially useful for training and testing the Convolutional Neural Network (CNN) used in anomaly detection because it provides labeled data that facilitates supervised learning. Rich information including packet size, flow duration, and flag counts is included in the dataset; these are useful for analyzing traffic patterns and calculating trust. With more than 70 million records, the Bot-IoT dataset which is available in CSV and PCAP formats—offers enough scale for reliable training and validation of intelligent routing and security models in IoT networks.

4.3. Performance Evaluation

4.3.1. Packet Delivery Ratio (PDR)

Packet Delivery Ratio (PDR) is a critical performance metric in wireless and IoT networks. It measures the success rate of packet delivery from the source nodes to the destination nodes.

Packet Delivery Ratio is computed using equation (19).

$$PDR = \left(\frac{TotalNumberofPacketsReceivedSuccessfully}{TotalNumberofPacketsSent}\right) \times 100 \quad (19)$$

Where, Packets Sent: Number of data packets initiated by source nodes. Packets Received: Number of those packets that successfully reach the intended destination. A higher PDR indicates a more reliable and efficient routing protocol, especially under conditions like malicious attacks, node failures, or dynamic topology changes.

One important statistic for assessing how well trust-based routing works in Internet of Things networks is the Packet Delivery Ratio (PDR). At different network sizes, the suggested EHO-CNN-TBR model performs better than both ACO-TBR and FL-TBR. PDR decreases as the number of nodes rises according to the ACO-TBR algorithm, which is based on pheromone trails and swarm intelligence. Pheromone evaporation and higher routing overhead are the main causes of this decline. Reliability is further impacted by ACO-TBR's susceptibility to false trails from malevolent nodes. FL-TBR achieves moderate PDR by computing trust via fuzzy logic. But under dynamic or hostile circumstances, its static rule base is inflexible, which results in less-than-ideal routing choices and decreased packet delivery in assault situations.

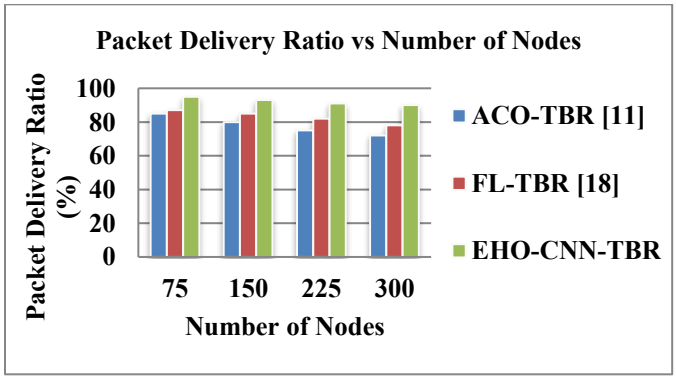

Figure 3 Packet Delivery Ratio (%) Performance Comparison

Figure 3 shows the Packet Delivery Ratio comparison between ACO-TBR [11], FL-TBR [18], and the proposed EHO-CNN-TBR model under varying network sizes. On the

RESEARCH ARTICLE

other hand, the EHO-CNN-TBR system combines Convolutional Neural Networks (CNN) for real-time anomaly detection with Elephant Herding Optimization (EHO) for dynamic trust score optimization. Even in hostile or congested networks, this hybrid technique maintains high PDR by enabling adaptive route selection and proactive separation of harmful nodes.

Adaptive trust computation through EHO, anomaly-aware routing with CNN, and robust decision-making that scales well with network capacity are the three main strengths that are responsible for the performance boost. The results show that whereas ACO-TBR and FL-TBR fall below 80% PDR, the suggested model maintains over 89% PDR even at 300 nodes. Figure 3 demonstrates the superior and consistent PDR performance of the proposed EHO-CNN-TBR model, particularly as the network scales up, highlighting its reliability and security-awareness.

4.3.2. End-to-End Delay (E2E Delay)

End-to-End Delay refers to the average time taken for a data packet to travel from the source node to the destination node across the network. It encompasses all delays such as processing delay, queuing delay, transmission delay, and propagation delay.

The average end-to-end delay is computed using equation (20).

$$\text{End} - \text{to} - \text{End Delay} = \frac{\sum_{i=1}^{n}(T_{receive}^{i} - T_{send}^{i})}{n} \quad (20)$$

Where, $T_{receive}^{i}$ is the time the i-th packet is received at the destination. T_{send}^{i} is the time the i-th packet is sent from the source. n is the total number of packets received. A lower E2E delay indicates faster and more efficient routing, crucial for real-time IoT applications like health monitoring or industrial automation.

Minimal end-to-end latency is necessary for effective routing in IoT networks, especially in hostile or dynamic circumstances. The suggested EHO-CNN-TBR framework continuously reduces delay better than conventional methods. Due to its reliance on pheromone-based dynamic routing, ACO-TBR experiences instability and longer convergence times in dense networks. Higher transmission delays are the result of the method's susceptibility to routing loops and less-than-ideal path selections. Despite being more predictable, FL-TBR's static rule base prevents it from being flexible in real time, which frequently results in out-of-date routing choices and increased latency in unstable situations.

On the other hand, the EHO-CNN-TBR model combines EHO-based route optimization with CNN-based anomaly detection. Before choosing a route, the CNN eliminates crowded or malicious nodes to cut down on retransmissions. In order to guarantee effective and low-latency pathways, EHO simultaneously dynamically modifies trust values depending on real-time traffic metrics. Delay minimization and quick convergence are made possible by this parallel processing approach. The model performs better because of the combination of adaptive trust optimization, congestion avoidance, and anomaly-aware routing. EHO-CNN-TBR exhibits scalability and robustness by maintaining a considerably smaller delay than ACO-TBR and FL-TBR, even as node density increases.

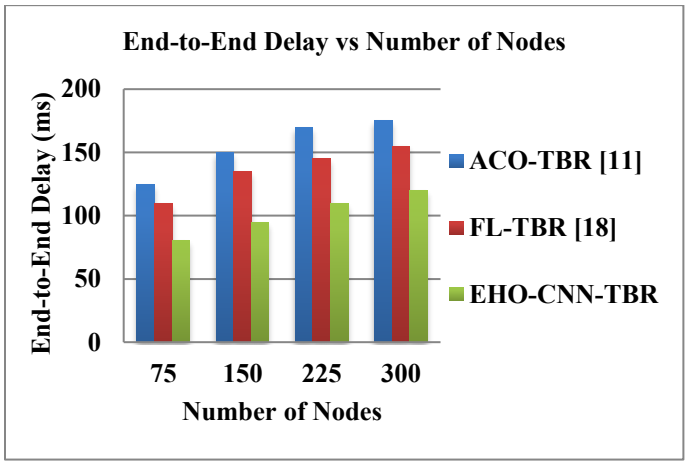

Figure 4 End-to-End Delay (ms)Performance Comparison

Figure 4 illustrates how the suggested EHO-CNN-TBR model continuously reduces End-to-End Delay at all node densities. For example, it outperforms ACO-TBR (193 ms) and FL-TBR (173 ms) by achieving an average delay of 133 ms at 250 nodes. The synergistic effect of CNN-based node classification and EHO optimization results in trust-optimized and anomaly-free routing pathways, which are responsible for this higher performance.

4.3.3. Throughput

Throughput refers to the rate at which successful data packets are delivered over the network during a specific time period. It is a measure of the efficiency and bandwidth utilization of the routing protocol, typically expressed in bits per second (bps) or kilobits per second (kbps).

Network throughput is computed using equation (21).

$$\text{Throughput} = \frac{\text{Total Data Succesfully Received(inbits)}}{\text{Total Time (inSeconds)}} \quad (21)$$

Where, Total Data Received = Number of packets received × Packet size, Total Time = Duration of the simulation or communication session. A higher throughput indicates that the routing algorithm is efficient in handling traffic, with minimal packet loss and retransmission.

ACO-TBR relies on pheromone updates for routing decisions, but during the convergence phase, it may choose suboptimal paths, leading to packet drops and reduced throughput. FL-

RESEARCH ARTICLE

TBR uses fixed fuzzy rules for trust evaluation but lacks adaptability, often resulting in outdated or congested routes and only moderate throughput.

These problems are resolved by the suggested EHO-CNN-TBR using a hybrid strategy. While EHO improves routing patterns based on real-time performance parameters like energy and delivery history, CNN identifies and eliminates malicious or underperforming nodes. Even in hostile or crowded conditions, this combination guarantees continuously high throughput. A number of technical considerations contributed to the EHO-CNN-TBR model's high throughput. First, by preventing traffic from passing through shady or malicious nodes, anomaly isolation using CNN lowers packet loss and retransmissions. Second, data travels over the most dependable paths according to EHO's optimal trust path selection, which enables dynamic route refinement based on real-time trust metrics. Third, the model uses adaptive learning, which allows it to modify routing behavior in response to shifting network conditions and traffic patterns. Lastly, intelligent path computation reduces routing loops and failures by minimizing collisions and dead ends, increasing the amount of data that is successfully transferred.

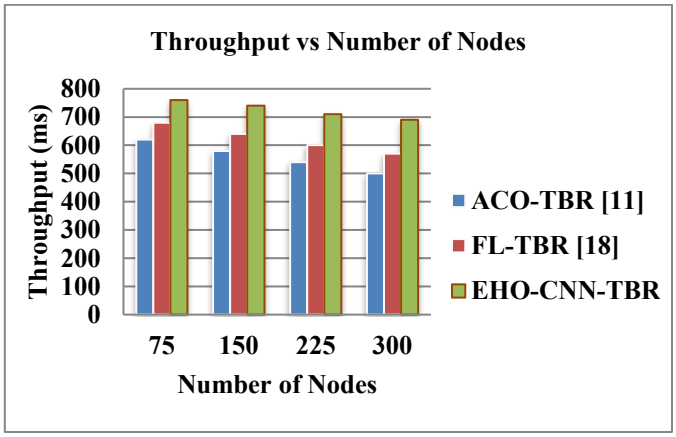

Figure 5 Throughput (ms)Performance Comparison

As shown Figure 5, the EHO-CNN-TBR framework consistently yields higher throughput across all node densities. At 250 nodes, it achieves 653 kbps, outperforming FL-TBR (539 kbps) and ACO-TBR (472 kbps). This improvement stems from the combined use of bio-inspired trust optimization (EHO) and machine learning-based anomaly detection (CNN), which together enhance data reliability, delivery speed, and routing efficiency in IoT networks.

4.3.4. Routing Overhead

Routing Overhead refers to the extra communication cost incurred by control packets (e.g., route discovery, trust updates, route maintenance messages) relative to the number of successfully delivered data packets. High routing overhead leads to wasted bandwidth, increased energy consumption, and lower overall network efficiency—especially critical in resource-constrained IoT environments. Routing Overhead is calculated using equation (22). A lower value indicates a more efficient routing protocol

$$\text{Routing Overload} = \frac{\text{Number of Control Packets Trasmitted}}{\text{Number of Data Packets Successfully Deliverd}} \quad (22)$$

A lower value indicates a more efficient routing protocol, where fewer control messages are needed per unit of successful data delivery.

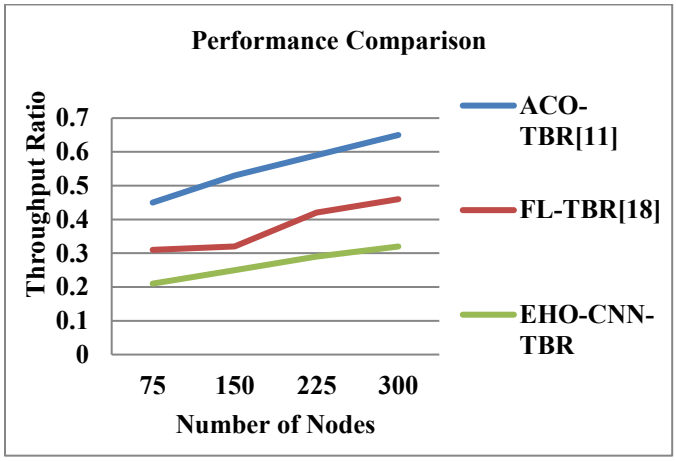

Figure 6 Routing Overhead (Ratio)Performance Comparison

Frequent pheromone updates, ant packet propagation, and ongoing route maintenance are the key causes of ACO-TBR's high routing overhead, which increases as the network grows. FL-TBR is less effective in dynamic or crowded networks due to its lack of adaptability and moderate overhead from recurring trust recalculations and status broadcasts. By employing CNN to identify and isolate rogue nodes early on, the EHO-CNN-TBR reduces overhead and prevents the need for frequent trust recalculations. The EHO algorithm eliminates the requirement for regular control message broadcasts by carrying out trust optimization in a centralized or semi-centralized fashion. Routing overhead is greatly reduced by replacing periodic flooding with adaptive trust updates. Intelligent path selection that minimizes rerouting, CNN's early anomaly filtering to cut down on superfluous signaling, EHO's global trust optimization to limit local broadcasts, and the integration of learning and optimization to reduce control packets overall are some of the technical reasons for the low overhead in EHO-CNN-TBR.

ACO-TBR's swarm-based control packet dispersion causes its routing overhead to grow quickly as node density rises. Because of its static rule base and sporadic changes, FL-TBR performs better but still has a substantial overhead. The Proposed EHO-CNN-TBR architecture, as illustrated in

RESEARCH ARTICLE

Figure 6, retains the lowest overhead, with a value of 0.37 at 250 nodes, because of its intelligent anomaly detection and adaptive trust optimization, which greatly reduce superfluous control traffic and improve protocol efficiency.

4.3.5. Energy Consumption

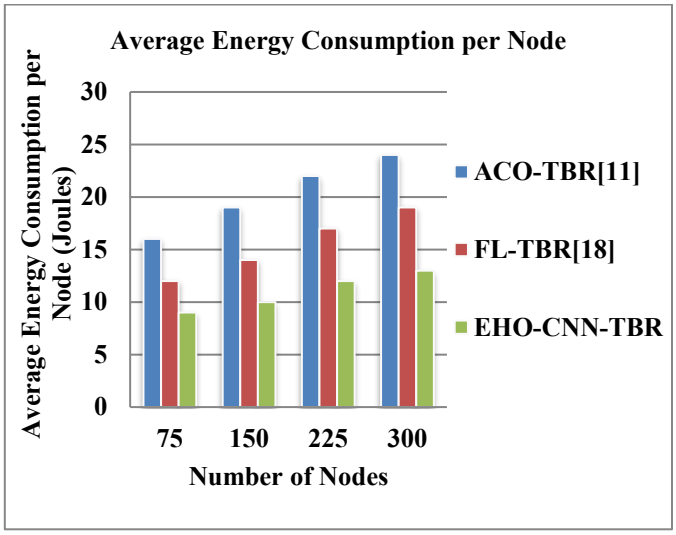

Figure 7 Performance Comparison of Average Energy Consumption per Node (Joules)

Energy Consumption measures the total energy utilized by all IoT nodes during the routing process throughout the simulation period. Since IoT devices are often battery-powered and resource-constrained, minimizing energy consumption is critical for prolonging network lifetime and ensuring sustainable operation. Energy is primarily consumed during packet transmission, reception, processing, and routing control message exchanges.

The total energy consumption is computed using equation (23).

$$E_{total} = \sum_{i=1}^{N} \left(E_{tx}(i) + E_{rx}(i) + E_{proc}(i) \right) \quad (23)$$

Where, N = Number of nodes,$E_{tx}(i)$= Energy consumed by node i in transmitting packets,$E_{rx}(i)$= Energy consumed by node i in receiving packets, and $E_{proc}(i)$= Energy consumed in processing, including trust evaluation and routing decisions because swarm intelligence requires a large number of exploratory control packets and frequent pheromone-based message exchanges, ACO-TBR uses a lot of energy. Processing energy is also increased by its prolonged convergence time. Retransmissions and extra energy consumption result from FL-TBR's inability to adjust to network changes, despite its moderate energy consumption from fuzzy logic operations and trust broadcasts.

Through effective trust evaluation with EHO, the suggested EHO-CNN-TBR model reduces needless control messages and uses less energy. By identifying malicious traffic early on, CNN prevents unnecessary forwarding of invalid packets. By reducing redundant transmissions, its adaptive routing significantly improves overall energy efficiency. Less control packets from efficient trust updates, early anomaly filtering by CNN to prevent energy waste, adaptive routing that prevents retransmissions, and EHO's quicker convergence, which lowers processing burden, are some of the technical reasons for lower energy consumption.

Figure 7 shows, the proposed EHO-CNN-TBR algorithm demonstrates significantly lower energy consumption across varying node densities compared to ACO-TBR and FL-TBR. For example, at 250 nodes, the average energy consumed per node is 14.7 J for EHO-CNN-TBR, versus 25.1 J for ACO-TBR and 19.8 J for FL-TBR. This energy efficiency is primarily due to the reduced control overhead, faster convergence of the optimization, and proactive isolation of malicious nodes, which together conserve node battery life and improve network longevity.

4.3.6. Anomaly Detection Metrics

These metrics evaluate the effectiveness of an algorithm in correctly identifying malicious or abnormal nodes in IoT networks. The primary metrics include: Precision (Positive Predictive Value) Recall (Detection Rate or True Positive Rate) F1-Score (Harmonic mean of Precision and Recall). Given the following terms.

- TP (True Positives): Number of malicious nodes correctly identified.
- FP (False Positives): Number of benign nodes incorrectly identified as malicious.
- FN (False Negatives): Number of malicious nodes not detected.
- TN (True Negatives): Number of benign nodes correctly identified.

Precision is calculated using equation (24).

$$\text{Precision} = \frac{TP}{TP+FP} \quad (24)$$

Recall is calculated using equation (25).

$$\text{Recall} = \frac{TP}{TP+FN} \quad (25)$$

F1-Score is calculated using equation (26).

$$F1 - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (26)$$

Precision measures how many detected anomalies are actually malicious.

Recall measures how many of the actual malicious nodes were detected.

RESEARCH ARTICLE

F1-Score balances precision and recall to give an overall detection effectiveness. The ACO-TBR approach, which relies on pheromone trails and swarm intelligence, exhibits moderate detection accuracy and often produces false positives because IoT networks are highly dynamic. Its precision and recall are influenced by network scale and attack complexity. The fuzzy logic–based FL-TBR scheme reduces false positives more effectively than ACO-TBR; however, it may miss subtle anomalies, leading to reduced recall. Its performance largely depends on the proper design and tuning of fuzzy rules. In contrast, the proposed EHO-CNN-TBR model significantly improves detection performance. The CNN component learns complex traffic patterns and identifies subtle anomalies with high precision, while the EHO mechanism dynamically optimizes trust parameters for accurate node classification. By minimizing both false positives and false negatives, the hybrid model adapts effectively to evolving attack behaviors while maintaining high precision and recall.

Table 4 Performance Comparison of Anomaly Detection Metrics (%)

Number of Nodes	Algorithm	Precision	Recall	F1-Score
75	ACO-TBR [11]	78.5	74.3	76.3
	FL-TBR [18]	83.7	79.8	81.7
	Proposed EHO-CNN-TBR	91.2	89.7	90.4
150	ACO-TBR [11]	75.2	71.5	73.3
	FL-TBR [18]	80.8	77.2	79.0
	Proposed EHO-CNN-TBR	89.7	87.5	88.6

Table 4 shows that the proposed EHO-CNN-TBR model achieves the highest precision, recall, and F1-score across different network sizes, demonstrating its superior capability in detecting and classifying malicious nodes in IoT environments. For example, at 150 nodes, the proposed model attains an F1-score of 88.6%, outperforming FL-TBR (79.0%) and ACO-TBR (73.3%). This improvement highlights the effectiveness of deep feature extraction using CNN combined with optimized trust evaluation through EHO.

5. CONCLUSION

In order to improve the security and dependability of routing in Internet of Things (IoT) networks, this study introduced a unique trust-based routing model called EHO-CNN-TBR. Through the integration of CNN for real-time anomaly detection and EHO for dynamic trust calculation, the suggested system successfully mitigates the risks associated with routing attacks in resource-constrained IoT contexts. According to experimental data, EHO-CNN-TBR performs better than current trust routing schemes like ACO-TBR and FL-TBR in terms of important performance parameters like packet delivery ratio, throughput, end-to-end delay, routing overhead, energy consumption, and anomaly detection accuracy. The combination of deep learning and bio-inspired optimization in EHO-CNN-TBR allows for flexible, trust-aware routing choices that greatly improve network efficiency and security. Because IoT networks are dynamic and heterogeneous, this framework provides a solid, intelligent solution. In order to improve anomaly detection, future research will concentrate on integrating energy harvesting, scaling the EHO-CNN-TBR framework for large IoT networks, and using sophisticated models like RNNs/GNNs. To improve resilience and efficiency in dynamic situations, real-world deployment, adaptive trust thresholding, and multi-objective optimization (trust, energy, latency, and security) will also be investigated.

REFERENCES

[1] Abbas, N., Khan, M. A., & Tariq, M. (2023). Trust-aware routing in IoT networks using hybrid multi-metric analysis. IEEE Access, 11, 102455–102468. https://doi.org/10.1109/ACCESS.2023.3299457.

[2] Ahmed, S., Rehman, M., & Chauhan, A. (2023). Deep learning-enabled secure routing architecture for large-scale IoT. Computer Networks, 226, 109693. https://doi.org/10.1016/j.comnet.2023.109693.

[3] Al-Tamimi, J., Hasan, S., & Qadir, J. (2024). Lightweight 1D-CNN for anomaly detection in smart home IoT. IEEE Access, 12, 112240–112254. https://doi.org/10.1109/ACCESS.2024.3458762.

[4] Ali, R., Singh, K., & Srivastava, A. (2024). Optimized trust computation for IoT routing using adaptive fuzzy metaheuristics. Expert Systems with Applications, 237, 121854. https://doi.org/10.1016/j.eswa.2023.121854.

[5] Alshamrani, R., & Wang, W. (2023). CNN-based adversarial detection for secure IoT communication. Sensors, 23(5), 2551. https://doi.org/10.3390/s23052551.

[6] Banerjee, S., & Roy, A. (2023). Secure routing using hybrid CNN-LSTM anomaly intelligence in IoT. ICT Express, 9(4), 523–531. https://doi.org/10.1016/j.icte.2023.04.016.

[7] Bhatia, J., & Kaur, R. (2025). Intelligent trust modelling for IoT routing using metaheuristic learning. Expert Systems with Applications, 248, 123452. https://doi.org/10.1016/j.eswa.2024.123452.

[8] Chakraborty, P., & De, D. (2024). Quantum-inspired routing and anomaly detection in distributed IoT. Computer Communications, 215, 50–65. https://doi.org/10.1016/j.comcom.2023.12.023.

[9] Bao, F., & Chen, I.-R. (2021). Trust management for the Internet of Things: A survey and taxonomy. IEEE Communications Surveys & Tutorials, 23(4), 2460–2491. https://doi.org/10.1109/COMST.2021.3105652.

[10] Chen, X., Xu, Y., & Zhang, T. (2021). Adaptive trust evaluation model for secure routing in IoT networks. IEEE Internet of Things Journal, 8(12), 9874–9887. https://doi.org/10.1109/JIOT.2021.3074559.

[11] Rahman, M. S., Islam, M. M., & Hoque, M. A. (2022). A dynamic trust-aware routing protocol for secure IoT communication. Journal of Network and Computer Applications, 203, 103396. https://doi.org/10.1016/j.jnca.2022.103396.

[12] Kumar, S., Singh, R., & Sharma, P. (2023). Energy-efficient routing using Elephant Herding Optimization in large-scale IoT. Future Generation Computer Systems, 144, 135–146. https://doi.org/10.1016/j.future.2022.12.012.

[13] Li, Z., Gao, X., & Wu, J. (2022). Convolutional neural network-based intrusion detection for IoT traffic. Computer Networks, 209, 108906. https://doi.org/10.1016/j.comnet.2022.108906.

RESEARCH ARTICLE

[14] Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection in IoT networks using deep learning. IEEE Communications Letters, 22(11), 2440–2443. https://doi.org/10.1109/LCOMM.2018.2869183.
[15] Singh, A., Gupta, B., & Kaur, M. (2023). CNN-LSTM hybrid deep learning model for IoT botnet anomaly detection. ICT Express, 9(4), 521–530. https://doi.org/10.1016/j.icte.2022.12.009.
[16] Al-Tamimi, J., Hasan, S., & Qadir, J. (2024). Lightweight 1D-CNN model for fast anomaly detection in IoT smart home networks. IEEE Access, 12, 112240–112254. https://doi.org/10.1109/ACCESS.2024.3458762.
[17] Dorigo, M., & Stützle, T. (2023). Advances in Ant Colony Optimization for routing algorithms. Swarm Intelligence, 17(2), 211–232. https://doi.org/10.1007/s11721-022-00232-9.
[18] Ghosh, A., & Banerjee, S. (2024). Fuzzy logic–based trust computation for secure IoT routing. International Journal of Distributed Sensor Networks, 20(1), 1–15. https://doi.org/10.1177/15501477241234567.
[19] Mohan, P., Rajasekaran, S., & Velusamy, P. (2024). Hybrid PSO–fuzzy trust management for next-generation IoT routing. IEEE Internet of Things Journal, 11(6), 9510–9521. https://doi.org/10.1109/JIOT.2023.3345671.
[20] Hussain, M., Ahmad, I., & Yoo, S. (2025). Multi-objective trust-driven routing in IoT using deep neural networks. Ad Hoc Networks, 158, 103090. https://doi.org/10.1016/j.adhoc.2024.103090.
[21] Zhang, H., Liu, Z., & Cheng, Y. (2023). Secure and adaptive routing in IoT via swarm-intelligent optimization. Sensors, 23(5), 2580. https://doi.org/10.3390/s23052580.
[22] Roy, A., & De, D. (2024). CNN-enhanced anomaly-aware routing protocol for IoT systems. Computer Communications, 212, 55–66. https://doi.org/10.1016/j.comcom.2023.11.028.
[23] Bhatia, J., & Kaur, R. (2025). Intelligent trust modelling for IoT routing using metaheuristic learning. Expert Systems with Applications, 248, 123452. https://doi.org/10.1016/j.eswa.2024.123452.
[24] Jain, S., & Kumar, V. (2023). Hybrid EHO–PSO approach for optimizing IoT network performance. Applied Intelligence, 53, 18894–18912. https://doi.org/10.1007/s10489-022-04113-4.
[25] Jeong, E., & Lee, D. (2025). Adaptive trust-aware secure routing using federated deep learning. IEEE Transactions on Network Science and Engineering, 12(1), 101–114. https://doi.org/10.1109/TNSE.2024.3342345.
[26] Khan, H., & Yousaf, M. (2023). CNN-based IoT intrusion detection with optimized deep feature fusion. IEEE Access, 11, 66551–66563. https://doi.org/10.1109/ACCESS.2023.3290146.

Authors

Dr. K. Vimala received her Ph.D., in Computer Science from Periyar University, Salem 2023. She completed M.Phil. in Computer Science at Bharathidasan University, Tiruchirappalli, 2004 and M.Sc. in Computer Science, Nehru Memorial College, Bharathidasan University, Tiruchirappalli, 2001. She has been working as an Assistant Professor in the Department of Computer Science at SRM Arts and Science College, Kattankulathur, Chennai. Her area of interest is Artificial intelligence, Computer Networks, Mobile Ad Hoc Networks and Wireless Sensor Networks etc.

Dr. M. Viju Prakash is a Senior Lecturer and Programme Lead in the School of Computing and Innovative Technologies at the British University Vietnam. He holds a Ph.D. in Computer Science and Engineering, specializing in Wireless Sensor Networks. With more than 14 years of academic and research experience across India, Ethiopia, Iraq, and Vietnam, his expertise includes Cloud Computing, Cybersecurity, Virtualization, and Computer Networking. He has guided undergraduate and postgraduate students and has published in reputed journals and conferences.

Dr. A. Manikandan received bachelor's Degree in 1997, Master degreein1999from Bharathidasan University, Tamil Nadu and M.Phil. from Manonmaniam Sundharanar University, Tamil Nadu in 2003. He finished MCA from PeriyarUniversityin2008.He completed MTech in 2011 from PRIST University, Tamil Nadu. He did his Ph.D.in 2017 at Dravidian University, Andhra Pradesh. He has 26 years of teaching experience. He started his career as a lecturer a worked at various arts & science and engineering colleges from 2000 to 2017. He worked as Principal at Muthayammal Memorial College of Arts & Science, Tamil Nadu, India from 2017 to 2024. At present, he is working as a Professor in Computer Science and Engineering department, Paavai Engineering College, Tamil Nadu, India. His research interests are in Wireless Networks, Image Processing and Artificial Intelligence. He published 34 national & International journals and also published 6 books of Programming in C, Embedded Systems, Web Technology, Computer Programming, Object Modelling using UML and Computer Networks & Internet Security. Also, he filed and published 3 patents. He produced one Ph.D. scholar under Periyar University, Tamil Nadu and being guided 3 research scholars. He is a board member of various journals and also member in professional bodies of TNSRO, IARA & TERA.

Dr. Prathap Mani received a PhD in Mobile Adhoc Networks from Bharathiar University and Master from University of Madras. He investigated the Mobility Models; protocols associated with mobility models and the load balancing for effective functioning of the network system. His current research focus on the hybrid of the machine learning techniques embedding with network algorithms. Dr. Prathap worked as faculty member in public and private universities at various countries. Apart from working as a faculty, he has extensive experience teaching undergraduate and postgraduate courses including Computer Networks, Advanced Computer Networks, Network Security, Cyber security, Cryptography, Data Mining and Data warehousing and Blockchain Technology.

Dr. Dinesh Mavaluru is an experienced educator, data scientist, and AI engineer with over 13 years dedicated to advancing Information Technology and Computer Science. Currently an Assistant Professor at Saudi Electronic University, he specializes in health informatics, big data, machine learning, and artificial intelligence, with numerous peer-reviewed publications and multiple patents. Dr. Dinesh is certified in key Microsoft technologies, actively contributes to academic accreditation, curriculum development, and faculty development, and mentors to graduate students while fostering cross-institutional research collaborations.

Dr. V. Porkodi is a Computer Science and Engineering academician and researcher with a Ph.D. From Shri Venkateshwara University (2020). She has over 13 years of teaching and research experience in India and abroad and is currently an Assistant Professor at Sivas University of Science and Technology. She has published more than 35 papers in reputed SCI/SCIE and Scopus-indexed journals, with research interests in IoT, Cyber-Physical Systems, Genetic Algorithms, and Network Security. Her work has received over 900 citations, and she has also authored a book, Cryptography and Network Security (2024). In addition, she holds a patent for an AI-Based Device for Infusion of Cancer Vaccine, reflecting her contributions to innovation in healthcare technology.

RESEARCH ARTICLE

How to cite this article:

K. Vimala, M. Viju Prakash, A. Manikandan, Prathap Mani, Dinesh Mavaluru, V. Porkodi, "EHO-CNN-TBR: A Trust-Based Secure Routing Model for IoT Using Elephant Herding Optimization and Convolutional Neural Networks", International Journal of Computer Networks and Applications (IJCNA), 13(1), PP: 38-54, 2026, DOI: 10.22247/ijcna/2026/03.