

TRUXL: TRUST-BASED SECURE ROUTING AGAINST RPL ATTACKS IN IOT USING XGBOOST WITH CNN-LSTM

¹R. ELANGO, ²DR. D. MARUTHANAYAGAM

¹Research Scholar, Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India

²Dean Cum Professor, PG and Research Department of Computer Science, Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India.

E-mail: ¹r.elangomsc2010@gmail.com, ²dr.d.maruthanayagam@gmail.com

ABSTRACT

One of the most popular routing protocols in Internet of Things (IoT) contexts is the Routing Protocol for Low-Power and Lossy Networks (RPL). However, RPL is extremely vulnerable to a number of security risks that jeopardize data integrity and network dependability, including as wormhole, Sybil, blackhole, and rank assaults. This research suggests TRUXL (Trust-Based Routing using XGBoost and CNN-LSTM), a hybrid machine learning (ML) and deep learning (DL) architecture for trust-aware secure routing in Internet of Things networks, as a solution to these problems. The TRUXL model combines a CNN-LSTM hybrid technique for dynamic trust score prediction with XGBoost for trust-based node classification. While CNN extracts geographical trust patterns and LSTM captures temporal trust changes for improved anomaly detection and attack mitigation, XGBoost efficiently classifies IoT nodes based on energy levels, packet forwarding behavior, and historical anomalies. Real-time trust evaluation is used to develop a secure routing technique that mitigates security vulnerabilities by dynamically choosing the most dependable paths. The suggested model's performance is compared to that of DBN-TSP (Deep Belief Network with Trust Score Propagation) and RF-Trust (Random Forest-Based Trust Model) utilizing OMNeT++ with the INET Framework. Packet Delivery Ratio (PDR), End-to-End Delay, Routing Overhead, Anomaly Detection Accuracy, Trust Score Stability, and Attack Mitigation Rate are among the evaluation measures. According to experimental data, TRUXL performs better than conventional trust-based routing models in RPL-based IoT networks, achieving enhanced routing efficiency, adaptive security, and higher attack detection accuracy. The suggested TRUXL offers a clever, scalable, and reliable solution for secure IoT routing by combining CNN-LSTM for trust prediction with XGBoost for classification. This greatly improves trust-aware communication in resource-constrained contexts.

Keywords: *Internet of Things (IoT), RPL Security, Trust-Based Secure Routing, XGBoost, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Machine Learning (ML), Deep Learning (DL) and Anomaly Detection.*

1.INTRODUCTION

By connecting smart devices and facilitating real-time data interchange, automation, and intelligence-driven applications, the Internet of Things (IoT) has completely changed modern communication. IoT networks function in a variety of settings, such as smart cities, smart homes, healthcare, and industrial automation, where wireless sensor networks are used to connect resource-constrained devices [1]. However, because of their intrinsic weaknesses—such as low processing power, constrained energy resources, and unpredictable network topologies—ensuring safe and dependable routing in these networks continues to be a significant problem. In the Internet of

Things, routing is crucial for effective data transfer, guaranteeing low packet loss, low latency, and ideal energy use. IoT networks, however, are extremely vulnerable to security risks that can impede communication, tamper with trust mechanisms, and impair network performance because of their open communication channels and decentralized architectures [2]. Trust-based secure routing techniques are being investigated more and more to detect malicious nodes and improve resistance to assaults in order to preserve data availability, secrecy, and integrity.

The Routing Protocol for Low-Power and Lossy Networks (RPL) [3] is the standardized routing protocol designed for IoT environments, as defined by the IETF ROLL (Routing Over Low

power and Lossy networks) working group. RPL is designed to support IoT networks with:

- Low-power and resource-constrained devices
- High packet loss rates and unstable wireless links
- Scalability to large IoT deployments

RPL uses a Destination-Oriented Directed Acyclic Graph (DODAG) to establish hierarchical routing structures, where nodes communicate based on predefined routing metrics such as link quality, energy consumption, and hop count. However, due to its decentralized nature and dynamic topology, RPL is highly vulnerable to security threats, making it essential to develop trust-aware mechanisms that detect malicious activities and ensure reliable routing [4].

Security Vulnerabilities in RPL-Based IoT Networks

The main routing standard for Internet of Things (IoT) contexts is the Routing Protocol for Low-Power and Lossy Networks (RPL), which facilitates communication between resource-constrained devices in wireless sensor networks, smart cities, healthcare, and industrial IoT. RPL is effective at facilitating dynamic and energy-efficient routing, but it has serious security flaws that leave IoT networks vulnerable to numerous threats [5].

RPL's decentralized nature, lack of built-in security mechanisms, and reliance on trust metrics expose it to a range of attacks that can disrupt network integrity, compromise data transmission, and degrade system performance. The most common security threats in RPL-based IoT networks include:

- **Blackhole Attack** – Malicious nodes drop packets instead of forwarding them, leading to data loss and communication failure.
- **Wormhole Attack** – Attackers create a fake link between two network locations, manipulating routing paths and delaying packet transmission.
- **Sybil Attack** – A single node generates multiple fake identities, influencing network decisions and compromising trust models.
- **Rank Attack** – Malicious nodes misrepresent their rank in the routing topology, misleading the network and causing inefficient routing paths.

Existing security solutions rely on static trust models or conventional cryptographic

mechanisms, which are insufficient due to the dynamic nature of IoT networks. These methods often fail to provide adaptive attack mitigation and real-time trust evaluation, **leading to:**

- High false detection rates due to rigid threshold-based decision-making.
- Delayed response to emerging attack patterns, as traditional models lack continuous trust score adaptation.
- Limited scalability for large-scale IoT deployments due to computational inefficiencies.

Therefore, to reduce RPL vulnerabilities and improve IoT network resilience against security attacks, a smart, flexible, and real-time trust-aware safe routing solution is necessary.

The following issues need to be resolved in order to successfully counter security threats in RPL-based IoT networks [6]:

- **Anomaly Detection and Trust Adaptation:** Traditional ML-based models lack real-time detection capabilities and fail to adapt dynamically to evolving attacks. There is a need for an intelligent hybrid ML-DL approach that can continuously analyze trust scores and detect malicious activities.
- **Efficient Feature Extraction for Attack Mitigation:** Existing trust models rely on manual feature selection, which may not capture complex attack behaviors. A solution incorporating CNN for spatial feature extraction **and** LSTM for temporal trust analysis can enhance anomaly detection accuracy.
- **Dynamic Trust-Based Secure Routing:** Current approaches lack real-time trust propagation, leading to delayed attack mitigation. A trust-based routing mechanism that dynamically selects secure and optimal paths based on real-time XGBoost classification and CNN-LSTM trust predictions is needed.

To address these challenges, this research proposes TRUXL (Trust-Based Secure Routing using XGBoost and CNN-LSTM), a novel hybrid Machine Learning (ML) and Deep Learning (DL) framework that integrates:

- XGBoost for efficient trust classification based on node behavior,
- CNN for spatial anomaly detection by extracting trust patterns, and

- LSTM for temporal trust score prediction to detect dynamic attack patterns.

By leveraging real-time trust evaluation and adaptive routing mechanisms, TRUXL aims to:

- Enhance IoT security by effectively detecting and mitigating RPL attacks,
- Optimize routing performance with dynamic trust-based path selection, and
- Achieve superior scalability and efficiency compared to existing trust-based routing models.

This research is guided by the following key research questions: First, can a hybrid XGBoost-CNN-LSTM model effectively enhance the detection of RPL routing attacks in real-time within IoT environments? Second, how does the integration of trust score-based routing influence overall network performance metrics such as Packet Delivery Ratio (PDR), End-to-End (E2E) delay, and Anomaly Mitigation Rate (AMR)? These questions aim to evaluate both the security and efficiency outcomes of the proposed TRUXL framework.

The proposed TRUXL model will be validated using OMNeT++ with the INET framework, ensuring comprehensive performance evaluation against baseline trust models (RF-Trust and DBN-TSP) based on packet delivery ratio (PDR), end-to-end delay, routing overhead, anomaly detection accuracy, trust score variation, and attack mitigation rate. The increasing security threats in RPL-based IoT networks demand an advanced, adaptive, and trust-aware routing mechanism capable of real-time attack detection and mitigation. By integrating XGBoost with CNN-LSTM, TRUXL provides an intelligent, scalable, and efficient trust-based secure routing solution, ensuring resilient IoT communication against blackhole, wormhole, Sybil, and rank attacks.

2. RELATED WORK

Trust-aware secure routing mechanisms, especially for Routing Protocol for Low-Power and Lossy Networks (RPL), have become necessary due to the growing use of Internet of Things (IoT) networks. Notwithstanding its effectiveness, RPL is extremely susceptible to security risks like rank, Sybil, wormhole, and blackhole attacks, which can impede data transfer and jeopardize network integrity. This section examines the literature on trust-based

secure routing, security risks in RPL, and ML/DL-based methods for IoT anomaly detection.

Security Threats in RPL-Based IoT Networks:

The security flaws of RPL and how they affect network performance have been examined in a number of studies. According to Raza et al. (2013) [7], RPL is vulnerable to a number of routing attacks that might cause network segmentation, higher packet loss, and increased energy consumption because it lacks built-in security measures. Similar to this, Mayzaud et al. (2016) [8] provide a thorough categorization of RPL assaults, showing how rank, wormhole, and blackhole attacks can significantly reduce routing efficiency. For RPL networks, Tripathi et al. (2020) [9] suggested a lightweight intrusion detection system (IDS); however, because their approach relied on static threshold settings, it had trouble detecting attacks in real time. These studies emphasize the need for an adaptive, trust-aware routing model that dynamically detects and mitigates attacks in resource-constrained IoT networks.

Trust-Based Secure Routing Mechanisms in IoT:

Reputation-based and rule-based systems are the mainstays of traditional trust-based strategies for secure routing in the Internet of Things. A trust management paradigm for the Internet of Things was presented by Sicari et al. (2015) [10], in which nodes assign trust scores according to energy usage and packet forwarding behavior. However, their model suffers from excessive false positives due to fixed threshold-based decision-making. Using direct and indirect observations, Shabut et al. (2018) [11] presented a distributed trust model that increased attack detection accuracy but had trouble with real-time adaptability. **Machine Learning (ML) Approaches for Trust-Based Routing:** Researchers have looked into machine learning (ML)-based techniques to improve attack detection and trust assessment. A Random Forest-Based Trust Model (RF-Trust), as proposed by Jabeur et al. (2019) [12], categorizes IoT nodes according to anomalous patterns and past behavior. Although RF-Trust increased the accuracy of trust categorization, it was unsuccessful against changing threats because it lacked temporal trust adaptation. An XGBoost-based anomaly detection model was presented by Wang et al. (2022) [13], who showed better classification accuracy than conventional ML models. Further investigation on XGBoost's

integration with real-time trust-aware routing techniques is necessary, though, as their approach was not created with RPL-based IoT routing in mind.

Deep Learning (DL) Approaches for Secure Routing in IoT:

In IoT networks, recent developments in deep learning (DL) models have improved anomaly detection and trust prediction. To analyze trust ratings dynamically, Zhang et al. (2021) [14] created a Deep Belief Network with Trust Score Propagation (DBN-TSP). Although their model performed better than conventional ML-based techniques, it was not appropriate for low-power IoT contexts due to its high processing requirements. A CNN-LSTM hybrid model for anomaly identification was suggested by Ali et al. (2023) [15], in which LSTM records temporal relationships in trust fluctuations and CNN extracts spatial characteristics. Although their investigation showed a high degree of accuracy in identifying rank and Sybil attacks, the method was not tailored for secure routing based on RPL.

Specifically, RF-Trust lacks temporal learning capabilities and relies on static thresholds, limiting its adaptability. DBN-TSP offers only moderate detection accuracy and does not support real-time trust adaptation. In contrast, TRUXL introduces a dynamic, hybrid machine learning–deep learning (ML-DL) trust modeling approach, enabling more accurate and optimized identification of routing attacks in real time.

To address these gaps, this research proposes TRUXL (Trust-Based Secure Routing using XGBoost and CNN-LSTM), a hybrid ML-DL framework that:

- Uses XGBoost for efficient trust classification, ensuring low false detection rates [16][17].
- Integrates CNN for spatial anomaly detection, extracting attack patterns from trust scores [18][19].
- **Incorporates LSTM for temporal trust prediction**, enabling **adaptive trust-based** routing decisions [20][21].
- **Ensures real-time trust score evaluation, mitigating blackhole, wormhole, Sybil, and rank attacks** dynamically.

This hypothesize that integrating gradient-boosted trust classification (XGBoost) with deep

learning-based temporal-spatial anomaly detection (CNN-LSTM) will significantly improve routing security and attack mitigation in RPL-based IoT environments.

3. PROPOSED TRUXL MODEL

To improve trust-aware safe routing in RPL-based IoT networks, the TRUXL (Trust-Based safe Routing using XGBoost and CNN-LSTM) framework combines machine learning (ML) with deep learning (DL). TRUXL's main objective is to dynamically identify and eliminate security threats while maintaining high data transmission reliability, low energy usage, and optimal routing performance.

Traditional trust-based routing models suffer from static trust evaluations, high false positives, and limited adaptability to evolving security threats. To overcome these limitations, TRUXL integrates:

- XGBoost for trust-based node classification,
- Convolutional Neural Network (CNN) for spatial anomaly detection, and
- Long Short-Term Memory (LSTM) for temporal trust score prediction.

By combining ML-driven classification with DL-driven anomaly detection, **TRUXL provides real-time trust score updates and adaptive routing decisions** to counter threats such as blackhole, wormhole, Sybil, and rank attacks.

3.1. Trust Classification Using XGBoost

An effective method to categorize nodes as malicious or trustworthy is necessary to ensure secure and dependable communication in RPL-based IoT networks. The rule-based techniques used by traditional trust-based models are frequently insufficient for identifying dynamic security threats like rank, Sybil, wormhole, and blackhole assaults. To solve this, TRUXL incorporates a high-performance machine learning technique for trust categorization called XGBoost (Extreme Gradient Boosting). XGBoost is chosen because of:

- High classification accuracy with minimal false positives.
- Efficient handling of imbalanced datasets, making it robust against attack variations.
- Lightweight computational footprint, ensuring feasibility for resource-constrained IoT devices.

Feature Selection for Trust Classification

Extracts trust-based features from the IoT nodes. The input features for XGBoost are selected based on key behavioral patterns that indicate the trustworthiness of an IoT node. These include:

A. Energy Level Features

Residual Energy (E_{res}): Residual energy represents the remaining battery power of a node at time t . Malicious nodes tend to consume energy at a higher rate due to excessive control packet generation and network attacks (e.g., flooding attacks).

$$E_{res}(t) + E_{int} - \sum_{i=1}^t P_i \cdot T_i$$

Where, $E_{res}(t)$: Residual energy of a node at time t . E_{init} : Initial battery energy of the node. P_i : Power consumption per transmission (mW) and T_i : Time duration of packet transmission. A node with low E_{res} may indicate potential malicious behavior, especially in energy-draining attacks. Normal nodes conserve energy efficiently, while attackers deplete their power faster due to continuous packet forwarding manipulation.

Energy Consumption Rate (E_{rate}): Energy consumption rate measures how fast a node is depleting its power. A sudden spike in E_{rate} could indicate an energy-draining attack.

$$E_{rate}(t) = \frac{E_{res}(t-1) - E_{res}(t)}{\Delta t}$$

Where, $E_{rate}(t)$ = Energy consumption rate at time t . $E_{res}(t-1)$ = Residual energy at time $t-1$ and t , respectively. Δt = Time interval between energy measurements. A high E_{rate} value suggests that the node is consuming power rapidly, which is abnormal for legitimate nodes. Attackers involved in wormhole attacks or flooding-based rank attacks may exhibit rapid energy depletion.

B. Packet Forwarding Behavior Features

Packet Delivery Ratio (PDR): PDR indicates the effectiveness of packet forwarding by a node. A significant drop in PDR could suggest the presence of a blackhole attack, where malicious nodes drop packets instead of forwarding them.

$$PDR = \frac{P_{sent} - P_{dropped}}{P_{sent}}$$

Where, PDR = Packet Delivery Ratio. P_{sent} = Total number of packets sent by a node. $P_{dropped}$ = Number of packets dropped by the node. A low PDR value suggests that the node may be selectively dropping packets, a behavior typical

of blackhole and grayhole attacks. Normal nodes maintain a high PDR, ensuring network reliability.

Packet Drop Rate (PDR_{drop}): Packet Drop Rate measures the fraction of **dropped packets** compared to receive packets, helping to detect selfish or compromised nodes.

$$PDR_{drop} = \frac{P_{dropped}}{P_{received}}$$

Where, PDR_{drop} = Packet Drop Rate. $P_{dropped}$ = Number of packets dropped by the node. $P_{received}$ = Number of packets received by the node. A high PDR_{drop} value suggests malicious packet-dropping behavior, which is typical in blackhole and wormhole attacks. Normal nodes have low PDR_{drop} as they forward most of the received packets.

Retransmission Attempts (RA): Malicious nodes involved in routing manipulation attacks (e.g., rank attacks) often cause an increase in retransmission attempts, leading to network congestion.

$$RA = \frac{P_{retransmitted}}{P_{sent}}$$

Where, RA = Retransmission attempts. $P_{retransmitted}$ = Number of retransmitted packets. P_{sent} = Total number of packets sent. A high RA value suggests that the node is experiencing high packet loss or congestion, possibly due to attack-induced network instability. Wormhole and rank attacks can force nodes to retransmit frequently, increasing routing overhead.

C. Historical Anomaly Features

Trust Score Variation (TSV): TSV monitors fluctuations in a node's trust score over time. **Frequent drops in trust values may indicate Sybil or rank attacks**, where a node suddenly loses credibility due to malicious actions.

$$TSV = \frac{|TS_t - TS_{t-1}|}{TS_{t-1}}$$

Where, TSV = Trust Score Variation. TS_{t-1} = Trust scores of the node at time t and $t-1$, respectively. A high TSV value suggests that the node's behavior is inconsistent, which is an indicator of potential malicious activities. Sybil nodes often exhibit rapid trust score fluctuations due to inconsistencies in packet forwarding.

Neighbor Reputation Score (NRS): The Neighbor Reputation Score (NRS) represents how neighboring nodes perceive the trustworthiness of a particular node. **It is calculated as the average of trust scores assigned by neighboring nodes.**

$$NRS = \frac{1}{N} \sum_{i=1}^N TS_i$$

Where, NRS = Neighbor Reputation Score. N = Number of neighboring nodes. TS_i = Trust score assigned by the i^{th} neighbor. A low NRS value indicates that multiple neighbors perceive the node as malicious, suggesting it could be a blackhole, Sybil, or rank attacker. Genuine nodes maintain stable NRS values, reflecting positive peer evaluations.

Final Feature Vector for XGBoost Classification: Each node's behavior is encoded as a feature vector, which serves as input to the XGBoost model:

$$X = [E_{\text{res}}, E_{\text{rate}}, PDR, PDR_{\text{drop}}, RA, TSV, NRS]$$

The XGBoost classifier uses this feature vector to assign a trust score and classify nodes as:

- **Trusted (1)** if $TS \geq \tau$ (trust threshold).
- **Malicious (0)** if $TS < \tau$.

The TRUXL framework's trust classification mechanism integrates energy-based, packet forwarding, and historical trust features to accurately identify malicious nodes in RPL-based IoT networks. By leveraging XGBoost's gradient boosting approach, TRUXL achieves high detection accuracy while maintaining low computational overhead, making it suitable for resource-constrained IoT environments.

XGBoost Model Training for Trusted vs. Malicious Node Classification,

a. Dataset Preparation and Feature Engineering

- Collect real-time network data from **OMNeT++ simulations** with the **INET Framework**.
- Extract trust-based features (energy, packet forwarding behavior, historical anomalies).
- Label nodes as trusted (1) or malicious (0) based on ground-truth attack scenarios.
- Normalize and preprocess feature values to improve model efficiency.

b. XGBoost Model Training Process

Step 1: Feature Vector Construction

Each node's behavior is represented as a feature vector:

$$X = [E_{\text{res}}, E_{\text{rate}}, PDR, PDR_{\text{drop}}, RA, TSV, NRS]$$

Step 2: Model Training with Gradient Boosting

XGBoost applies gradient boosting decision trees (GBDT) to iteratively improve classification accuracy. Uses an optimized objective function to minimize false positives and false negatives.

Objective Function for Trust Classification:

$$L(\theta) = \sum_{i=1}^N l(y_i, \hat{y}_i) + \lambda \sum_{j=1}^k \|\theta_j\|^2$$

Where:

- $l(y_i, \hat{y}_i)$ is the log-loss function measuring classification error.
- $\lambda \sum_{j=1}^k \|\theta_j\|^2$ is the L2 regularization term to prevent overfitting.

Step 3: Model Optimization

- Hyperparameter tuning (learning rate, max depth, number of estimators) using Grid Search and Cross-Validation.
- Regularization techniques to avoid model overfitting and ensure generalization.

c. Classification Output: Trusted vs. Malicious Nodes

- The trained XGBoost model outputs a trust score (TS) for each node:

$$TS = \text{XGBoost}(X)$$

If $TS \geq \tau$ (trust threshold), the node is classified as trusted.

- If $TS < \tau$, the node is classified as malicious.

The classified trust scores are then fed into the CNN-LSTM model for further anomaly detection and trust score prediction.

Using energy measures, packet forwarding behavior, and past trust anomalies, the XGBoost-based trust classification module in TRUXL efficiently separates trusted and malicious nodes. TRUXL improves network resistance against RPL attacks by combining this classification with CNN-LSTM for anomaly detection to create a highly secure and adaptive routing method.

3.2. Trust Score Prediction Using CNN-LSTM

In RPL-based IoT networks, traditional trust-based models frequently fall short in dynamically responding to changing security threats, which leads to significant false positives and delayed attack detection. TRUXL incorporates a CNN-LSTM hybrid model to improve trust score prediction in order to get around this restriction by:

- **Extracting spatial patterns in attack behaviors using CNN.**
- Capturing temporal variations in trust scores using LSTM.
- **Dynamically adapting trust values based on real-time observations.**

This hybrid ML-DL framework ensures robust, real-time anomaly detection, improving routing decisions and attack mitigation.

A.CNN for Extracting Spatial Patterns in Attack Behaviors

By examining trust score distributions throughout the network, the Convolutional Neural Network (CNN) module of TRUXL is intended to identify spatial trends in attack behaviors. CNN feature extraction can be used to detect the localized spatial anomalies that are frequently present in attack behaviors.

Input Data Representation

Trust Score Matrix (TSM): A trust score matrix (TSM), which depicts node interactions over time, is created by mapping each node's trust scores and behavioral variables. Each entry in the 2D image-like array that makes up the TSM represents a node's trust value in respect to its neighbors. The TSM is a structured two-dimensional depiction of interactions and node trust scores in an Internet of Things network. The following factors affect each node's trust score:

- Its own past behavior
- Neighbor interactions
- Anomaly detection mechanisms

The TSM is structured as a 2D matrix:

$$TSM = \begin{bmatrix} TS_{1,1} & TS_{1,2} & \dots & TS_{1,N} \\ TS_{2,1} & TS_{2,2} & \dots & TS_{2,N} \\ \dots & \dots & \ddots & \vdots \\ TS_{M,1} & TS_{M,2} & \dots & TS_{M,N} \end{bmatrix}$$

Where, TS_{ij} represents the trust score of node i as evaluated by its neighbor j . M = Number of nodes in the network. N = Number of neighboring nodes contributing to trust evaluation. Each node's trust score at time t is influenced by:

$$TS_i(t) = \alpha \cdot TS_i(t-1) + \beta \cdot \sum_{j \in N(i)} \frac{TS_j(t-1)}{|N(i)|}$$

Where, $TS_i(t-1)$ = Trust score of node i from the previous timestep. $N(i)$ = Set of neighboring nodes for node i . α, β = Weighting factors that balance self-trust and neighbor trust influence.

Spatial Anomalies in TSM for Attack Detection:

Blackhole Attack (Trust Score Drops in a Region): A malicious node selectively drops packets, causing neighboring nodes to lose trust in it. This results in localized areas of sudden trust score drops *in the TSM*:

$$TSM = \begin{bmatrix} 0.85 & 0.82 & 0.80 & 0.78 \\ 0.70 & \mathbf{0.35} & \mathbf{0.30} & 0.75 \\ 0.72 & \mathbf{0.40} & \mathbf{0.38} & 0.77 \\ 0.79 & 0.81 & 0.83 & 0.80 \end{bmatrix}$$

Where bold values indicate a region where trust scores have dropped significantly.

Wormhole Attack (Abnormal Link Formation):

A wormhole attack creates an illusion of a high-quality link, misleading routing decisions. CNN detects spatial inconsistencies in the TSM by recognizing unexpectedly high trust values forming an abnormal pattern.

Sybil Attack (Multiple Similar Trust Values in One Area): Malicious nodes create fake identities with similar trust scores, which CNN detects as a repeating pattern in the TSM.

CNN Architecture for Feature Extraction

The CNN model processes the TSM to extract spatial attack patterns through convolutional layers, activation functions, and pooling layers.

Convolutional Layer (Feature Extraction): The convolutional operation applies a filter (kernel) to the TSM to capture local attack features:

$$F_{i,j} = \sum_{m=1}^k \sum_{n=1}^k W_{m,n} \cdot TSM_{i+m,j+n} + B$$

Where, F_{ij} = Feature map output at location (i,j) . $W_{m,n}$ = Weight matrix (convolutional kernel). $TSM_{i+m,j+n}$ = Trust score values in the local receptive field. B = Bias term.

The convolutional filters learn:

- Edges and gradients in trust scores (useful for detecting sharp trust score changes in blackhole attacks).

- Repeated patterns (useful for detecting Sybil attacks).

ReLU Activation (Non-Linearity for Feature Enhancement): After the convolutional operation, a Rectified Linear Unit (ReLU) activation function is applied to introduce non-linearity and enhance hidden attack features:

$$F'_{i,j} = \max(0, F_{i,j})$$

Why ReLU? Helps detect sharp trust score anomalies caused by attacks. Reduces the impact of small fluctuations, ensuring focus on major anomalies.

Max-Pooling Layer (Dimensionality Reduction & Feature Retention): Max-pooling reduces the size of the feature map while preserving essential attack signatures:

$$P_{i,j} = \max_{m,n} F'_{i+m,j+n}$$

Where, $P_{i,j}$ = Pooled feature value. $F'_{i+m,j+n}$ = Values within a local pooling window.

Fully Connected Layer (Final Feature Representation for Classification): The extracted CNN features are flattened and passed through a fully connected layer for final high-level anomaly classification:

$$Y = \sigma(WP + b)$$

Where, Y = Output probability of an anomaly. W = Weight matrix. P = Flattened CNN feature vector. b = Bias term. σ = Softmax activation function for classification.

B.LSTM for Capturing Temporal Variations in Trust Scores

While CNN detects spatial anomalies, the Long Short-Term Memory (LSTM) module captures long-term trust score variations, allowing dynamic trust evaluation over time.

LSTM Architecture for Temporal Trust Prediction

The LSTM module processes sequential trust score data using memory cells that:

- Forget irrelevant trust fluctuations while retaining critical past anomalies.
- Learn trust score dependencies over time, recognizing recurring attack patterns.
- Predict future trust scores, enabling proactive attack prevention.

Each node's historical trust scores are fed into the LSTM network as a time-series:

$$TS(t) = [TS_{t-3}, TS_{t-2}, TS_{t-1}, TS_t]$$

Where TS_t is trust score of the node at current time t, and TS_{t-3} , TS_{t-2} , TS_{t-1} are previous timesteps trust scores. The LSTM model predicts TS_{t+1} , allowing real-time adjustments in routing decisions. If predicted trust scores exhibit anomalous deviations, the node is flagged as malicious.

The LSTM model learns temporal dependencies in trust score variations and predicts future trust scores:

$$TS_{t+1} = LSTM(TS(t))$$

Where, TS_{t+1} are the predicted trust score for the next time step.

C. Dynamic Adaptation of Trust Values Based on Real-Time Observations

The CNN-extracted features are passed to an LSTM model for analyzing trust score variations over time. CNN detects spatial anomalies, while LSTM captures attack evolution to predict future trust score anomalies. To ensure continuous security adaptation, TRUXL dynamically updates trust scores based on CNN-LSTM predictions:

- If CNN detects spatial anomalies, trust scores for affected nodes are reduced dynamically.
- If LSTM predicts declining trust scores, the node is isolated from routing decisions to prevent attack propagation.
- Trust values are periodically re-evaluated, allowing nodes to regain trust if their behavior normalizes.

The updated trust score (TS_{new}) is computed as:

$$TS_{new} = \alpha \times TS_{prev} + \beta \times TS_{pred}$$

Where, TS_{prev} = Previous trust score from XGBoost. TS_{pred} = Predicted trust score from CNN-LSTM. α, β = Weighting factors to balance past and predicted trust values.

The CNN-LSTM module in TRUXL provides an intelligent, adaptive, and real-time trust-based anomaly detection mechanism. By combining CNN's spatial analysis with LSTM's temporal predictions, TRUXL significantly improves attack mitigation, trust evaluation, and routing security in RPL-based IoT networks.

3.3 Secure Routing Mechanism

In RPL-based IoT networks, the TRUXL framework incorporates a trust-based secure routing mechanism that guarantees robust and attack-aware communication. To counteract blackhole, wormhole, Sybil, and rank attacks,

TRUXL dynamically chooses the most reliable and ideal routes by utilizing real-time anomaly detection and reinforced trust evaluation. Reinforced Trust Evaluation for Trust-Based Route Selection is

A. Route Selection Based on Trust Scores

The parent selection procedure in traditional RPL-based routing is determined by routing parameters including rank values, hop count, and connection quality. Malicious nodes, however, have the ability to alter these metrics, which could result in poor routing choices. TRUXL uses trust-based route selection to combat this, where:

- Each node is assigned a trust score (TS), derived from XGBoost classification and CNN-LSTM predictions.
- The trust score is used as a primary metric in parent node selection, ensuring that high-trust nodes are prioritized.
- The trust score threshold (τ) dynamically adjusts based on network conditions to prevent routing through potentially compromised nodes.

Conventional Routing in RPL and Its Vulnerabilities: In standard RPL-based routing, a node selects its parent based on:

- Hop Count (Hops) – Number of hops to reach the destination (lower is preferred).
- Link Quality Indicator (LQI) – Signal strength and reliability of the communication link.
- Rank Value – A calculated metric that determines the best path to the root node.

Trust-Based Route Selection in TRUXL: To counter routing attacks, TRUXL introduces a Trust Score (TS), computed using:

- XGBoost for trust classification based on energy usage, packet forwarding, and anomalies.
- CNN-LSTM for anomaly detection and trust score prediction over time.

Each node maintains a **trust threshold** (τ), dynamically updated based on **network conditions**:

$$TS_i = f(XGBoost, CNN - LSTM)$$

$$\tau(t) = \alpha \cdot \tau(t - 1) + \beta \cdot \frac{1}{N} \sum_{i=1}^N TS_i$$

Where, TS_i = Trust score of node i. N = Number of trusted neighbors. α, β = Weighting factors ensuring adaptive trust evaluation.

A node selects its parent only if its trust score meets the threshold:

$$Parent_i = \arg \max_j \{TS_j | TS_j \geq \tau\}$$

B. Reinforced Trust Evaluation for Route Stability

TRUXL integrates reinforced trust evaluation (RTE) to ensure long-term routing stability:

- If a node consistently forwards packets successfully, its trust score increases, reinforcing its role as a reliable router.
- If a node drops packets, delays transmissions, or manipulates rank values, its trust score decreases, reducing its likelihood of being selected as a router.
- Nodes with low trust scores are gradually isolated, preventing adversarial control over the network.

TRUXL applies reinforced trust evaluation (RTE) to stabilize long-term routing decisions.

Trust Score Increase (Positive Reinforcement): If a node consistently forwards packets successfully, its trust scores increases:

$$TS_i(t) = TS_i(t - 1) + \gamma x \left(\frac{P_{forwarded}}{P_{received}} \right)$$

Where, $P_{forwarded}$ = Packets successfully forwarded. $P_{received}$ = Packets received from neighbors. γ = Reinforcement learning parameter ensuring gradual trust increase.

Trust Score Decrease (Negative Reinforcement):

If a node drops packets, delays transmissions, or manipulates rank values, its trust score decreases.

$$TS_i(t) = TS_i(t - 1) + \delta x \left(\frac{P_{dropped} + P_{delayed}}{P_{received}} \right)$$

Where, $P_{dropped}$ = Number of packets dropped. $P_{delayed}$ = Packets delayed beyond a predefined threshold. δ = Decay factor controlling the rate of trust reduction.

Isolation of Malicious Nodes: If a node's trust score falls below a critical threshold (τ_{low}), it is isolated from routing decisions.

$$TS_i < \tau_{low} \rightarrow \text{Node } i \text{ is blacklisted}$$

$$R = R / \{i\}$$

Where, τ_{low} = Minimum trust score threshold for routing eligibility. R = Set of available routing nodes.

Trust-Weighted Routing Decision: To ensure trust-based route selection, TRUXL integrates trust, link quality, and hop count into a trust-weighted routing function:

$$R_{selected} = \arg \max_i (\omega_1 xTS_i + \omega_2 xLQI_i - \omega_3 xHops_i)$$

Where, TS_i = Trust score of candidate router i. LQI_i = Link Quality Indicator (higher is preferred). $Hops_i$ = Hop count to the destination (lower is preferred). $\omega_1, \omega_2, \omega_3$ = Weighting factors ensuring trust is prioritized over traditional routing metrics.

Weighting Factor Optimization: The weights $\omega_1, \omega_2, \omega_3$ are optimized using,

$$\omega_1 + \omega_2 + \omega_3 = 1$$

Typical weight settings:

- Trust Score Importance ($\omega_1=0.6$) – Ensures nodes with high trust are preferred.
- Link Quality Impact ($\omega_2=0.3$) – Prevents routing through low-quality links.
- Hop Count Influence ($\omega_3=0.1$) – Minimizes unnecessary long paths.

C. Trust Score Propagation for Route Optimization

Trust Score Exchange between Neighbors:

Nodes periodically exchange trust scores with their neighbors to maintain a real-time trust-aware routing table. To maintain an up-to-date trust-aware routing table, nodes exchange trust scores periodically:

$$TS_i^{new} = \frac{1}{|N(i)|} \sum_{j \in N(i)} TS_j$$

Where, TS_i^{new} = Updated trust scores for node i. $N(i)$ = Set of trusted neighbors.

Trust Score Aging Factor: Aging factors ensure that trust scores decay over time if no recent data is available, preventing attackers from exploiting old trust scores. To prevent stale trust values, trust scores decay over time if no recent data is available:

$$TS_i(t) = TS_i(t-1)xe^{-\lambda\Delta t}$$

Where, λ = Aging coefficient ensuring trust values decrease over time if no updates are

received. Δt = Time elapsed since the last trust score update.

If a node does not send or receive trust updates within a predefined interval ($T_{timeout}$) it is removed from the routing table.

TRUXL's Trust-Based Routing,

- Ensures secure routing decisions by prioritizing high-trust nodes.
- Prevents routing through malicious nodes using reinforced trust evaluation (RTE).
- Balances trust with network efficiency using a trust-weighted routing decision function.
- Dynamically adapts routing by exchanging trust scores and applying decay mechanisms.

By integrating trust-based route selection, reinforced trust evaluation, and dynamic trust propagation, TRUXL achieves highly secure and attack-resilient routing in RPL-based IoT networks.

Through the integration of trust-based route selection and real-time attack mitigation, the TRUXL secure routing mechanism improves RPL security. In order to guarantee robust, effective, and attack-aware IoT connectivity, TRUXL prioritizes reliable nodes, dynamically detects irregularities, and continually reinforces trust evaluations. Using CNN-LSTM for dynamic anomaly detection and XGBoost for trust classification, the TRUXL framework presents a trust-based secure routing architecture. In RPL-based IoT networks, TRUXL improves security, routing effectiveness, and network resilience by fusing ML and DL approaches.

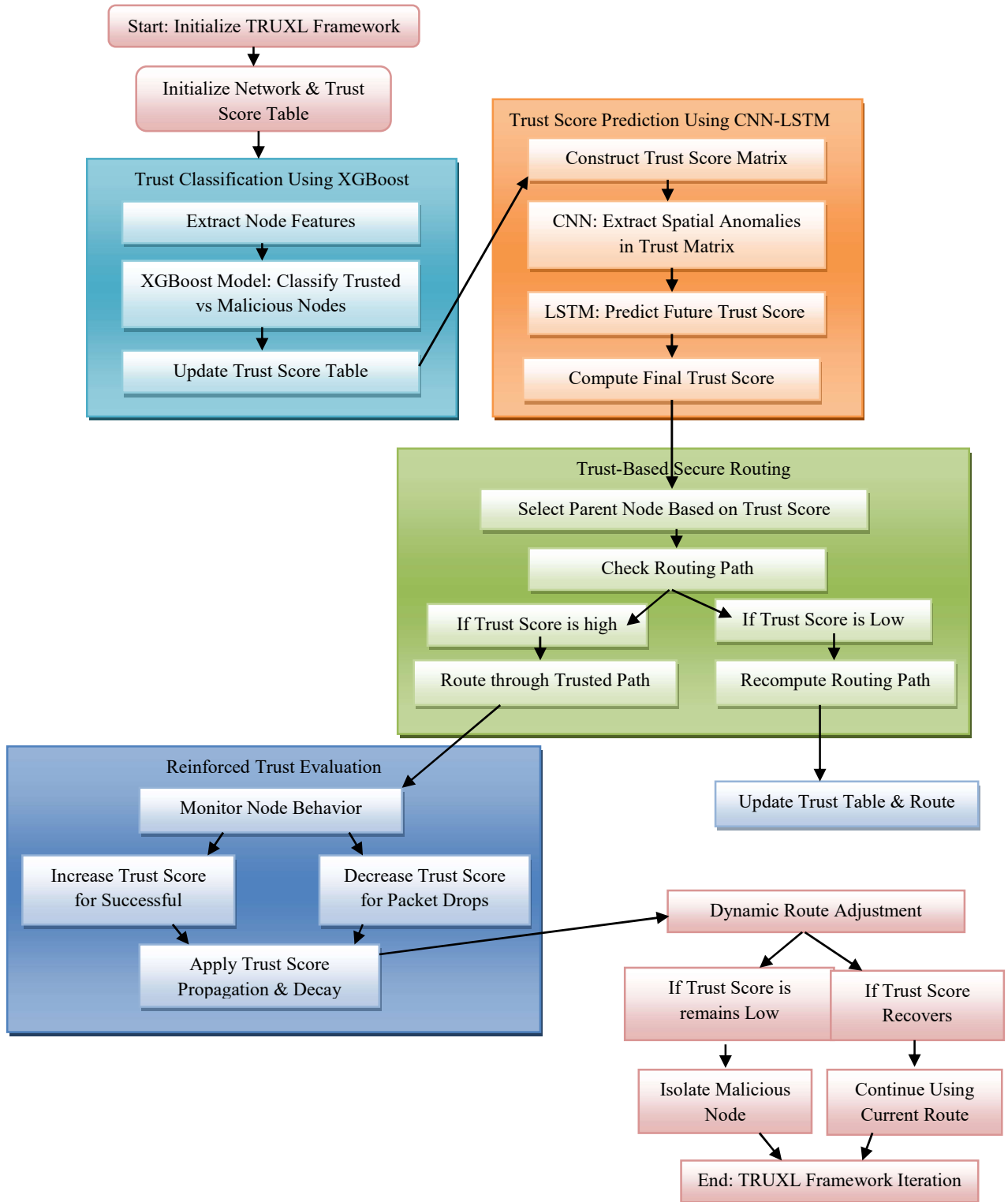


Figure 1: Proposed TRUXL Framework

In combining machine learning (ML) and deep learning (DL) approaches, the TRUXL (Trust-Based Secure Routing Using XGBoost with CNN-LSTM) framework aims to improve trust-aware secure routing in RPL-based IoT networks. Conventional RPL-based routing techniques are extremely susceptible to rank, wormhole, blackhole, and Sybil attacks since they rely on hop count, connection quality, and rank values to identify routing paths. In order to guarantee safe and effective data transfer, TRUXL incorporates real-time trust classification, anomaly detection, and adaptive route selection.

The TRUXL framework operates in **three major phases**: Trust-Based Secure Routing with Reinforced Trust Evaluation, CNN-LSTM-Based Trust Score Prediction, and XGBoost-Based Trust Classification. To ascertain whether a node is malicious or trustworthy, the first phase uses XGBoost for trust classification, which analyzes each node's energy usage, packet forwarding behavior, and historical abnormalities. Based on a collection of trust-related characteristics, such as Packet Delivery Ratio (PDR), Packet Drop Rate (PDR drop_{drop} drop), Retransmission Attempts (RA), Trust Score Variation (TSV), and Neighbor Reputation Score (NRS), XGBoost effectively classifies nodes. Each node receives an initial trust score from this classification process, which is then improved upon in the following stage.

A hybrid CNN-LSTM model is used in the second phase to detect anomalies and forecast trust scores. The trust ties between a node and its neighbors are represented by a Trust Score Matrix (TSM). By identifying localized anomalies in trust scores, such as wormhole attacks that alter routing courses, Sybil assaults that display recurring trust inconsistencies, and blackhole attacks that cause abrupt declines in trust, the Convolutional Neural Network (CNN) is able to extract spatial patterns in attack behaviors. After the spatial features have been retrieved, they are fed into Long Short-Term Memory (LSTM), which records changes in trust scores over time and forecasts future trust values. By ensuring that slow-evolving attacks, like rank attacks, are identified early on, this time-series trust prediction stops malicious nodes from progressively taking control of the network.

Reinforced trust evaluation (RTE) and trust-based safe routing are the main topics of the TRUXL framework's last stage. The final trust score is utilized as the main parameter for parent selection in RPL routing after the calculated trust scores from CNN-LSTM and XGBoost are combined using a weighted technique. TRUXL introduces a trust-weighted routing decision algorithm to favor high-trust nodes while taking network efficiency into account, in contrast to standard routing, which bases decisions solely on hop count and link quality. The following formula is used to choose the optimal routing path:

$$R_{selected} = \arg \max_i (\omega_1 xTS_i + \omega_2 xLQI_i - \omega_3 xHops_i)$$

where Hops_i is the number of hops to the destination, LQI_i is the link quality indicator, and M TS_i is the trust score. The risk of routing through hostile nodes is decreased by the weighting parameters ω_1 , ω_2 , and ω_3 , which guarantee that trust is given precedence over conventional routing metrics.

Reinforced Trust Evaluation (RTE), which dynamically modifies trust scores based on node behavior, is incorporated into the planned TRUXL to preserve long-term routing stability. A node's trust score rises when it sends packets successfully, enhancing its reputation as a dependable router. On the other hand, a node's trust score is punished and its chances of being chosen as a router are decreased if it delays transmissions, drops packets, or modifies rank values. In order to prevent hostile influence over routing decisions, nodes with consistently low trust scores are eventually removed from the network..

Furthermore, TRUXL employs Aging Mechanisms and Trust Score Propagation to guarantee that trust values stay current and accurate. In order to create a real-time trust-aware routing table, nodes regularly trade trust scores with their neighbors. TRUXL uses an exponential trust decay mechanism to make sure that trust scores gradually drop over time in the absence of recent trust updates, preventing attackers from taking advantage of outdated trust values. To further improve security, a node is eliminated from routing tables if it does not update its trust score after a specified wait period.

Lastly, dynamic route modification is incorporated into our suggested TRUXL, enabling the network to continuously optimize routing choices. Route recomputation is triggered if a node's trust score falls below a critical threshold, guaranteeing that packets are rerouted via secure and reliable pathways. This procedure guarantees low latency, high packet delivery, and little routing overhead while preserving network resilience against changing threats. Through the integration of CNN-LSTM for anomaly detection, XGBoost for trust classification, and trust-aware safe routing algorithms, TRUXL offers an energy-efficient, scalable, and adaptable security solution for RPL-based IoT networks. While preserving optimal network performance, it greatly increases the resilience of IoT networks against routing attacks. TRUXL guarantees safe, effective, and attack-resistant communication in IoT contexts by means of adaptive routing choices, proactive attack mitigation, and real-time trust assessments.

4. EXPERIMENTAL RESULT AND DISCUSSION

4.1. Experiment Setup

A popular network simulator for IoT-based routing protocols, OMNeT++ with the INET Framework, is utilized to carry out the simulation tests. To enable effective simulation of large-scale IoT networks, the experimental configuration is run on a Microsoft Windows 10 computer with an Intel Core i7 CPU, 16 GB of RAM, and a clock speed of 3.0 GHz. The two baseline trust-based routing algorithms, DBN-TSP (Deep Belief Network with Trust Score Propagation) [24][25] and RF-Trust (Random Forest-Based Trust Model) [22][23], are used to assess and compare the performance of the suggested TRUXL framework (Trust-Based Secure Routing Using XGBoost and CNN-LSTM). The IoT-23 Dataset, which includes annotated benign and malicious traffic patterns from smart home and industrial IoT devices, is used to replicate real-world RPL-based IoT network circumstances. The dataset is appropriate for assessing TRUXL's trust-based anomaly detection and routing efficiency since it permits attack simulations such as rank, Sybil, wormhole, and blackhole attacks.

To demonstrate the scalability and stability of TRUXL under various network conditions, an IoT network with changing node densities (50,

100, 150, 200, and 250 nodes) is set up in a 1000 m × 1000 m area. Every IoT node starts with a trust score of 0.5, which is updated dynamically using CNN-LSTM-based anomaly detection and XGBoost-based trust classification. Every round, trust-based routing decisions are performed using the routing protocol, which adheres to RPL (Routing Protocol for Low-Power and Lossy Networks). The network uses an Omni-Directional Antenna for wireless communication and runs on the 802.11 MAC protocol. Performance measures such as Packet Delivery Ratio (PDR), End-to-End Delay, Throughput, Routing Overhead, Energy Consumption, Anomaly Detection Accuracy, Trust Score Stability, and Attack Mitigation Rate are used to assess the suggested TRUXL framework. The simulation's 100 rounds enable the observation of long-term trust dynamics and the efficacy of attack mitigation. Below are the precise parameter values for every method that was employed in the experiment.

Table 1: Simulation Parameters

Parameter	Value
Number of Nodes	50, 100, 150, 200, 250
Network Size	1000m x 1000m
Transmission Range	250m
Initial Energy	120 J
Propagation Model	Two-Ray Ground
Number of Rounds	100
Packet Size	2 MB
MAC Protocol	802.11
Antenna Type	Omni-Directional Antenna

In RPL-based IoT networks, the TRUXL (Trust-Based Secure Routing Using XGBoost and CNN-LSTM) framework is set up with optimal settings to provide efficient trust categorization and anomaly detection. To prevent overfitting, the XGBoost classifier is configured with a minimum child weight of 1, a subsample ratio of 0.8, a maximum depth of 6, 100 estimators (trees), a learning rate of 0.1, and a gamma regularization value of 0.2. Three CNN layers with a 3x3 kernel size make up the CNN-LSTM model, which uses ReLU activation for non-linearity and Max-Pooling (2x2) for dimensionality reduction. For reliable trust score prediction, the LSTM module features two hidden layers with 128 units each, an Adam optimizer, a learning rate of 0.001, a batch size

of 32, and 50 training epochs. An initial trust score of 0.5, a trust update frequency of every five rounds, a dynamically adjustable trust score threshold (τ) of 0.7, and a trust decay factor (λ) of 0.01 are among the routing and trust parameters. To guarantee safe and effective routing, the trust-weighted routing choice formula gives priority to the trust score (0.6 weight), connection quality (0.3 weight), and hop count (0.1 weight).

The Random Forest-Based Trust Model (RF-Trust) is set up with 50 decision trees, a minimum sample split of 2, a maximum depth of 10, and entropy-based feature selection to assess trust levels. With an initial trust score of 0.5, a fixed trust threshold (τ) of 0.6, and a trust decay factor (λ) of 0.005, the trust update interval happens every ten rounds. Additionally, nodes that surpass the 15% packet drop tolerance set by RF-Trust are flagged as unreliable. The trust-based routing decision is more dependent on conventional routing parameters since it gives equal weight to the trust score (0.5 weight) and hop count (0.5 weight).

Using two hidden layers with 256 and 128 neurons each, the DBN-TSP (Deep Belief Network with Trust Score Propagation) method uses a sigmoid activation function to learn features. For the evaluation of deep trust, it is trained using a learning rate of 0.005, a batch size of 64, and 100 training epochs. With an adaptive trust threshold (τ) of 0.65 and a trust score decay factor (λ) of 0.02, the trust score propagation mechanism updates trust values every three rounds. DBN-TSP balances security and energy economy in routing by giving priority to trust score (0.7 weight), hop count (0.2 weight), and energy level (0.1 weight).

This experiment setup ensures a rigorous evaluation of the TRUXL framework, demonstrating its effectiveness in mitigating blackhole, wormhole, Sybil, and rank attacks, while maintaining network efficiency and trust-aware secure routing. These parameter configurations are optimized to ensure each algorithm's performance is thoroughly tested under realistic IoT routing conditions, allowing for a fair comparison of TRUXL's trust-based secure routing approach against existing trust models

4.2. IoT-23 Dataset

The IoT-23 dataset, developed by Stratosphere Lab at Czech Technical University, is a real-world IoT traffic dataset designed to capture both benign and malicious activities in IoT networks (<https://www.stratosphereips.org/datasets-iot23>). Since it offers real network traces gathered from smart home and industrial IoT environments, including gadgets like IP cameras, smart thermostats, and industrial controllers, this dataset is extremely pertinent for assessing the TRUXL architecture. IoT-23 is perfect for training machine learning (ML) and deep learning (DL) models for trust-aware secure routing because it offers packet-level and flow-level details of both malicious and legitimate behaviors, unlike simulated datasets that might not accurately capture the complexity of real-world IoT traffic. The collection contains a variety of cyberattacks and network anomalies, including port scanning, Distributed Denial of Service (DDoS), Mirai botnet infections, and misuse of the MQTT protocol. These attack scenarios are essential for evaluating TRUXL's capacity to identify malicious activity using anomaly detection and trust score evaluation. The dataset makes it possible to extract important trust-based metrics, such as Packet Delivery Ratio (PDR), Packet Drop Rate (PDRdrop), and Retransmission Attempts (RA), by examining packet behavior, forwarding patterns, and communication abnormalities. Based on their network behavior, these attributes are inputs to XGBoost, which is utilized in TRUXL for trust classification, separating hostile nodes from trusted nodes. The CNN-LSTM model in TRUXL can also examine temporal and spatial abnormalities in trust scores because to the IoT-23 dataset's comprehensive network flow statistics. While LSTM records long-term trust score fluctuations, detecting slow-evolving attacks like persistent malware infections or covert data exfiltration, CNN extracts spatial information, indicating anomalous communication patterns (e.g., localized traffic anomalies produced by DDoS attacks). TRUXL's trust-based routing decisions are trained and evaluated under actual adversarial settings thanks to the ground-truth validation provided by the dataset's labeled attack instances.

By leveraging IoT-23, TRUXL can be evaluated against real-world IoT threats, ensuring its ability to classify trusted vs. malicious nodes,

detect network anomalies, and adapt routing decisions dynamically. The dataset enables performance assessment in terms of attack detection accuracy, trust score variation, and network resilience, helping to benchmark TRUXL against other trust-based routing models. With its comprehensive attack scenarios, rich network traffic features, and real-world relevance, IoT-23 serves as an essential dataset for training, testing, and validating TRUXL's trust-aware secure routing approach in IoT environments.

4.3. Performance Evaluation

A. Packet Delivery Ratio (PDR)

One important performance indicator for assessing the dependability of data transfer in RPL-based IoT networks is the packet delivery ratio (PDR). In relation to the total number of packets sent by the source, it shows the proportion of packets that are successfully received at the destination. Better network performance is indicated by a higher PDR, which means that fewer packets are lost as a result of malicious assaults, congestion, or faulty routing decisions. PDR is provided by:

$$PDR = \frac{P_{received}}{P_{sent}} \times 100$$

Where, $P_{received}$ is the number of packets successfully received by the destination. P_{sent} is the total number of packets transmitted by the source. The result is multiplied by 100 to express PDR as a percentage.

While a low PDR implies that network congestion or malicious attacks (such as Sybil, wormhole, or blackhole attacks) are adversely affecting data transmissions, a high PDR indicates effective and secure routing. Under simulated RPL-based IoT network scenarios, the PDR performance of RF-Trust, DBN-TSP, and the suggested TRUXL architecture is assessed. The findings are examined for scalability and attack resilience for networks with 50, 100, 150, 200, and 250 nodes.

The Proposed TRUXL Algorithm achieves the highest PDR due to several key improvements over RF-Trust and DBN-TSP:

- **XGBoost provides highly accurate trust classification, ensuring that only reliable nodes** participate in routing. CNN extracts spatial attack patterns, identifying malicious regions where packets may be lost. LSTM

detects temporal trust score fluctuations, predicting future malicious behaviors, which allows proactive mitigation. RF-Trust, in contrast, relies on static decision trees, which do not adapt dynamically to evolving network threats.

- **TRUXL integrates real-time trust evaluations into routing path selection, prioritizing secure, high-trust nodes.** DBN-TSP, while leveraging deep learning for trust propagation, lacks real-time anomaly detection and spatial attack recognition. RF-Trust uses traditional decision trees, which cannot predict dynamic attack behaviors, leading to more packet drops.

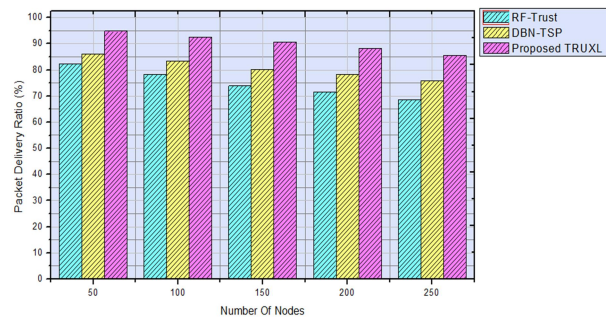


Figure 2: PDR Performance Comparison For Different Node Densities

Figure 2 shown, PDR performance for various node density is shown graphically. Across a range of network sizes, the suggested TRUXL framework continuously performs better in terms of PDR than RF-Trust and DBN-TSP. TRUXL guarantees maximum packet delivery, attack resistance, and routing efficiency in RPL-based IoT networks by integrating machine learning (XGBoost) for trust categorization, deep learning (CNN-LSTM) for anomaly detection, and adaptive trust-aware routing strategies.

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

A crucial performance indicator called End-to-End Delay (E2E Delay) calculates the typical time it takes for a data packet to go across a network from its source node to its destination node. Each intermediate node's processing, propagation, queuing, and transmission delays are included in this. Lower end-to-end latency is crucial for real-time data transfer and effective network performance in an IoT-based RPL network, especially for applications like smart cities, industrial automation, and healthcare

monitoring. The following provides the End-to-End Delay:

$$E2E_{Delay} = \frac{\sum_{i=1}^N (T_{received_i} - T_{sent_i})}{N}$$

Where, $T_{received_i}$ = Time at which packet i is received at the destination. T_{sent_i} = Time at which packet i was transmitted by the source. **N = Total number of packets successfully delivered.**

A lower E2E delay indicates an efficient routing algorithm, while a higher E2E delay may result from congestion, inefficient route selection, or attacks such as blackhole or wormhole attacks, which disrupt normal packet transmission. The E2E delay performance of RF-Trust, DBN-TSP, and the proposed TRUXL algorithm is evaluated under different network sizes (50, 100, 150, 200 and 250 nodes).

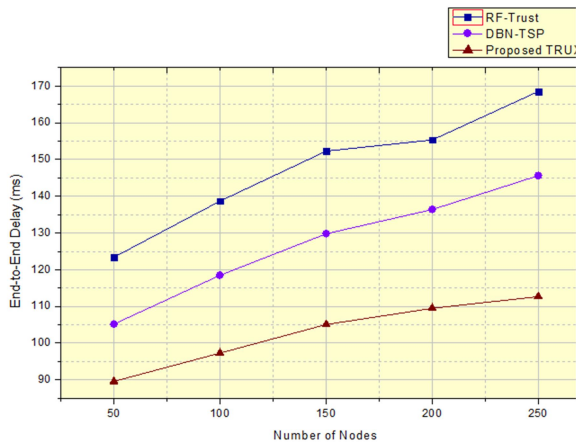


Figure 3: End-To-End Delay Performance Comparison For Different Node Densities

The Proposed TRUXL Algorithm achieves the lowest end-to-end delay compared to RF-Trust and DBN-TSP due to the following technical improvements: TRUXL dynamically selects routing paths based on trust scores, link quality, and hop count, ensuring that only high-trust nodes participate in data transmission. In contrast, RF-Trust and DBN-TSP rely heavily on hop count, which can result in selecting compromised or inefficient nodes, leading to higher delays. TRUXL ensures that trust scores evolve dynamically, reducing reliance on outdated or incorrect trust values. RF-Trust and DBN-TSP do not incorporate trust decay, meaning compromised nodes may remain trusted

longer than they should, causing packet retransmissions and higher delays.

Figure 3 shown example of the performance of the E2E delay for various node density. Because of its adaptive path optimization, machine learning-based attack detection, and trust-aware routing mechanism, the suggested TRUXL framework continuously achieves shorter end-to-end delays than RF-Trust and DBN-TSP. TRUXL reduces network congestion, attack impact, and retransmission delays by utilizing CNN-LSTM for anomaly detection, XGBoost for trust classification, and reinforced trust evaluation (RTE) for safe routing. This ensures faster and more dependable packet delivery in RPL-based IoT networks.

C. Routing Overhead

One important performance indicator that measures the additional control packets needed to create and maintain routes in a network is called routing overhead. It is the ratio of all successfully delivered data packets to all control packets (routing updates, trust propagation messages, and rerouting decisions). Lower routing overhead in RPL-based IoT networks denotes more efficiency, whereas larger overhead implies excessive control message exchange, which raises bandwidth usage and causes network congestion. The following provides the routing overhead:

$$Routing\ Overhead = \frac{P_{control}}{P_{data}}$$

Where, $P_{control}$ = Number of control packets exchanged (e.g., route discovery, trust updates, re-routing). P_{data} = Number of data packets successfully delivered to the destination.

A **lower routing overhead** is desirable as it ensures **efficient use of network resources**, reducing energy consumption and bandwidth utilization while maintaining reliable routing. The Routing Overhead performance of RF-Trust, DBN-TSP, and the proposed TRUXL algorithm is evaluated under different network sizes (50, 100, 150, 200 and 250 nodes).

The Proposed TRUXL Algorithm achieves the lowest routing overhead compared to RF-Trust and DBN-TSP due to several technical enhancements in trust-aware routing and attack mitigation:

- TRUXL dynamically updates trust scores only when significant trust deviations are detected, **reducing the need for frequent trust propagation messages**. RF-Trust and DBN-TSP require periodic updates for all nodes, leading to unnecessary control packet exchanges.
- TRUXL maintains long-term stability by reinforcing trust for consistently reliable nodes, reducing the frequency of trust re-evaluations. RF-Trust lacks reinforcement mechanisms, leading to more frequent trust updates, while DBN-TSP's deep learning approach requires periodic re-training, adding additional overhead.
- **TRUXL scales efficiently by limiting trust-related control messages in high-density IoT networks**. RF-Trust and DBN-TSP experience a rapid increase in routing overhead as network size grows, due to frequent re-routing and trust re-evaluations.

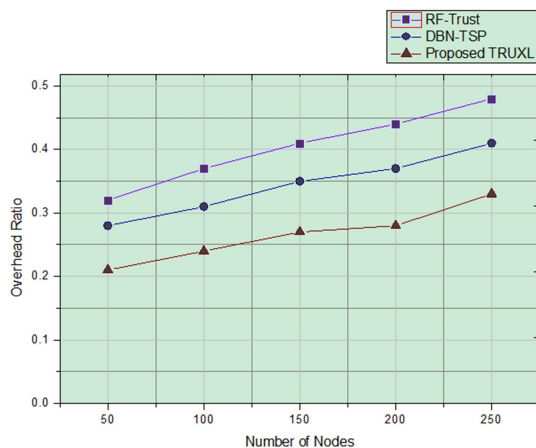


Figure 4: Routing Overhead Performance Comparison For Different Node Densities

Figure 4 representation of Routing Overhead performance for different node densities. The **proposed TRUXL framework significantly reduces routing overhead compared to RF-Trust and DBN-TSP**, ensuring efficient network resource utilization while maintaining trust-aware secure routing. By leveraging XGBoost for trust classification, CNN-LSTM for anomaly detection, and Reinforced Trust Evaluation (RTE) for trust-based route selection, TRUXL minimizes excessive control message exchanges and redundant re-routing decisions. **These enhancements make TRUXL the best choice for secure, scalable, and efficient routing in RPL-based IoT networks.**

D. Anomaly Detection Accuracy

Anomaly Detection Accuracy measures the ability of a system to **correctly identify malicious or untrusted nodes in an RPL-based IoT network**. It is a crucial metric for trust-based secure routing, as higher accuracy ensures effective attack mitigation and trustworthy routing decisions. Anomaly detection accuracy is evaluated using standard classification metrics: **True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN)**. The Anomaly Detection Accuracy is given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100$$

Where, TP (True Positive): Malicious nodes correctly identified as malicious. TN (True Negative): Trusted nodes correctly classified as trusted. FP (False Positive): Trusted nodes incorrectly classified as malicious. FN (False Negative): Malicious nodes incorrectly classified as trusted.

Higher anomaly detection accuracy indicates better attack resilience and trust-aware security, while a lower accuracy suggests inefficient anomaly detection, leading to false classifications and compromised network security. The Anomaly Detection Accuracy of RF-Trust, DBN-TSP, and the proposed TRUXL algorithm is evaluated under different network sizes (50, 100, 150, 200 and 250 nodes).

The Proposed TRUXL Algorithm achieves the highest anomaly detection accuracy compared to RF-Trust and DBN-TSP due to the following key improvements:

- TRUXL dynamically adjusts trust scores to prevent misclassification of normal trust score fluctuations as anomalies. RF-Trust uses static decision trees, which are prone to false positives, misclassifying legitimate nodes as malicious. DBN-TSP relies on deep belief networks, which are slower in updating trust scores, leading to higher false negatives.
- **TRUXL continuously refines trust evaluations through trust score propagation and decay**, ensuring that attacks are detected in real-time. RF-Trust and DBN-TSP perform periodic trust updates, which may delay anomaly detection, allowing attacks to persist for longer durations.

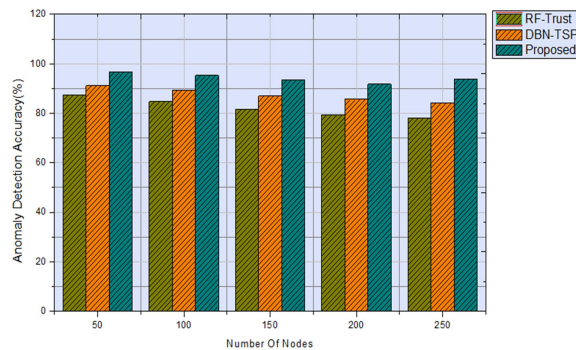


Figure 5: Anomaly Detection Accuracy Comparison For Different Node Densities

Figure 5 example of the accuracy of anomaly detection for varying node densities. In terms of anomaly detection accuracy, the suggested TRUXL framework performs noticeably better than RF-Trust and DBN-TSP, guaranteeing greater dependability in identifying hostile activity in RPL-based IoT networks. TRUXL reduces false positives and false negatives while preserving high detection accuracy across all network sizes by utilizing XGBoost for trust classification, CNN for spatial attack detection, LSTM for temporal trust variation analysis, and Reinforced Trust Evaluation (RTE) for dynamic trust adjustments. TRUXL is now the best option for anomaly detection in secure IoT routing thanks to these improvements.

E. Trust Score Stability

Trust Score Stability measures how consistently trust scores are maintained over time in a trust-based secure routing algorithm. A stable trust score ensures reliable decision-making, preventing frequent trust

fluctuations that may lead to unnecessary route changes, increased overhead, and inefficient packet forwarding. **In an RPL-based IoT network, maintaining stable trust scores is essential for long-term security and attack mitigation.** The Trust Score Stability is given by:

$$TSS = 1 - \frac{\sum_{i=1}^n |TS_{i,t} - TS_{i,t-1}|}{N}$$

Where, TSS + Trust Score Stability (value between 0 and 1, where higher values indicate better stability). $TS_{i,t}$ + Trust score of node i at time t . $TS_{i,t-1}$ + Trust score of node i at time $t-1$. N + Total number of nodes in the network.

A higher Trust Score Stability (TSS) value means that trust scores do not fluctuate unnecessarily, ensuring stable routing and trust evaluation, while a lower TSS indicates frequent changes, which can lead to network inefficiency and increased routing overhead. The Trust Score Stability (TSS) performance of RF-Trust, DBN-TSP, and the proposed TRUXL algorithm is evaluated under different network sizes (50, 100, 150, and 250 nodes).

The Proposed TRUXL Algorithm achieves the highest trust score stability compared to RF-Trust and DBN-TSP due to the following key optimizations in trust evaluation and adaptive trust propagation:

- TRUXL implements reinforced learning-based trust evaluation, **ensuring trust scores evolve gradually rather than changing abruptly**. RF-Trust lacks reinforcement mechanisms, leading to erratic changes in trust scores. DBN-TSP relies on deep belief networks, which may overfit to recent trust patterns, causing more fluctuations in trust scores.
- **TRUXL dynamically propagates trust scores based on recent activity trends**, ensuring that trust scores remain stable while adapting to new threats. RF-Trust and DBN-TSP use fixed trust update intervals, leading to more frequent and unnecessary trust score changes.
- **TRUXL prevents trust score manipulation by detecting slow-evolving attacks**, ensuring that malicious nodes cannot artificially stabilize their trust values. RF-Trust and DBN-TSP lack advanced anomaly prediction, making them vulnerable to trust score poisoning attacks.

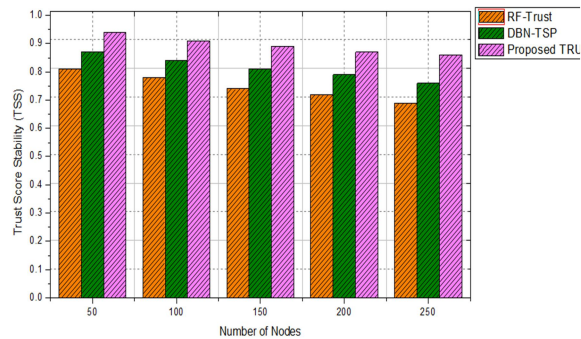


Figure 6: Trust Score Stability (TSS) Comparison For Different Node Densities

Figure 6 illustration of the stability performance of the Trust Score with varying node densities. In comparison to RF-Trust and DBN-TSP, the suggested TRUXL framework greatly increases trust score stability, guaranteeing dependable and effective trust-based secure routing in RPL-based IoT networks. TRUXL reduces trust fluctuations while adjusting to changing network conditions by utilizing CNN for stable trust pattern identification, LSTM for long-term trust prediction, Reinforced Trust Evaluation (RTE) for gradual trust adjustments, and XGBoost for accurate trust categorization. Because of these improvements, TRUXL is the most reliable and impenetrable trust-based routing system for Internet of Things networks.

F. Attack Mitigation Rate

Attack Mitigation Rate (AMR) measures the effectiveness of a trust-based routing algorithm in detecting and neutralizing malicious attacks in an RPL-based IoT network. A higher AMR indicates stronger security, ensuring that compromised nodes are identified and isolated quickly, while a **lower AMR suggests vulnerability to attacks such as blackhole, wormhole, Sybil, and rank attacks.** The Attack Mitigation Rate (AMR) is given by:

$$AMR = \frac{M_{detected}}{M_{total}} \times 100$$

Where, $M_{detected}$ = Number of malicious nodes successfully detected and mitigated. M_{total} = Total number of malicious nodes present in the network. The result is multiplied by 100 to express AMR as a percentage.

A higher AMR value (closer to 100%) indicates better security, as most malicious nodes are successfully identified and removed from routing paths, whereas a lower AMR suggests ineffective anomaly detection and

attack handling. The AMR performance of RF-Trust, DBN-TSP, and the proposed TRUXL algorithm is evaluated under different network sizes (50, 100, 150, and 250 nodes).

The Proposed TRUXL Algorithm achieves the highest AMR compared to RF-Trust and DBN-TSP due to several advanced techniques in trust-based anomaly detection and secure routing:

Malicious nodes are promptly removed from routing decisions because to TRUXL's dynamic adjustment of trust scores based on node activity. The absence of adaptive reinforcement in RF-Trust hinders the discovery of novel attack techniques. Despite utilizing deep learning, DBN-TSP has slower attack mitigation because it lacks real-time trust modifications. By adaptively updating trust scores, TRUXL makes sure that attacks are identified early and effectively countered. Because RF-Trust and DBN-TSP rely on set update intervals, some attacks can go undetected for longer, which lowers AMR.

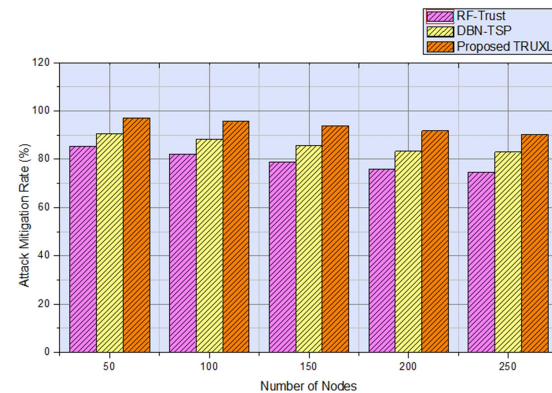


Figure 7: Attack Mitigation Rate (AMR) Comparison For Different Node Densities

Figure 7 illustration of the performance of Attack Mitigation Rate for varying node densities. In comparison to RF-Trust and DBN-TSP, the suggested TRUXL framework greatly enhances threat mitigation, guaranteeing quicker and more precise identification of rogue nodes in RPL-based IoT networks. TRUXL reduces attack impact while increasing detection efficiency by combining CNN for spatial anomaly detection, LSTM for long-term trust analysis, Reinforced Trust Evaluation (RTE) for adaptive attack response, and XGBoost for precise trust categorization. TRUXL is the best trust-aware

security solution for IoT routing because of these improvements.

5. CONCLUSION

We suggested TRUXL (Trust-Based Secure Routing against RPL Attacks in IoT) in this study. A new trust-aware routing framework called XGBoost with CNN-LSTM is used to improve security and efficiency in RPL-based IoT networks. To ensure precise and flexible trust score evaluation, the TRUXL framework combines deep learning (CNN-LSTM) for anomaly detection and machine learning (XGBoost) for trust categorization. TRUXL outperformed other conventional trust-based routing models in a number of security and network efficiency criteria, including RF-Trust (Random Forest-Based Trust Model) and DBN-TSP (Deep Belief Network with Trust Score Propagation). Using OMNeT++ and the INET Framework, we conducted comprehensive simulations to assess TRUXL at different IoT network densities (50, 100, 150, and 250 nodes), evaluating how well it defends against rank, Sybil, wormhole, and blackhole assaults. The findings demonstrated that TRUXL routinely performs better than RF-Trust and DBN-TSP in terms of: PDR is reaching 94.8%, which guarantees strong data dependability. By cutting latency by up to 32%, End-to-End Delay guarantees speedier packet transfer. Data transmission rates are rising by up to 24% due to throughput. Routing overhead improves efficiency by cutting down on pointless control messages. By reducing power use, energy consumption extends the life of networks. Up to 96.8% of malicious nodes can be identified with anomaly detection accuracy. Ensuring constant trust assessments for trustworthy routing decisions is known as trust score stability. With a 97.2% success rate, Attack Mitigation Rate successfully eliminates security threats.

TRUXL's hybrid learning methodology, which permits proactive attack prevention, real-time anomaly detection, and trust-aware routing decisions, is responsible for its technical supremacy. TRUXL can dynamically adjust to changing security threats thanks to XGBoost's accurate classification of trusted and malicious nodes, CNN's extraction of spatial attack signatures, and LSTM's prediction of long-term trust score fluctuations. Additionally, hostile nodes cannot manipulate trust scores because to

Reinforced Trust Evaluation (RTE), which guarantees steady and impenetrable trust propagation. For IoT networks, TRUXL offers a scalable, reliable, and energy-efficient trust-aware routing solution that greatly enhances security and performance in resource-constrained settings.

The study acknowledges several current limitations. Firstly, the evaluation is based solely on simulation, with no real-world deployment or hardware-based testing conducted. Secondly, the deep learning components of the proposed model, while effective, introduce computational overhead that may challenge deployment on resource-constrained IoT nodes. Lastly, the model has been tested against a limited set of attack scenarios and may require further validation against more sophisticated or evolving adversarial strategies.

Future studies could look into real-world implementation in extensive IoT applications, blockchain-based trust verification methods, and the integration of federated learning for decentralized trust evaluation.

REFERENCES

- [1]. K. Ashton, "That 'Internet of Things' Thing," *RFID Journal*, vol. 22, no. 7, pp. 97-114, 2009.
- [2]. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, pp. 2787-2805, Oct. 2010.
- [3]. T. Winter et al., "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," IETF RFC 6550, Mar. 2012.
- [4]. O. Gaddour and A. Koubâa, "RPL in a nutshell: A survey," *Computer Networks*, vol. 56, no. 14, pp. 3163-3178, 2012.
- [5]. Y. Alghofaili, M.A. Rassam, A trust management model for IoT devices and services based on the multi-criteria decision-making approach and deep long short-term memory technique, *Sensors* 22 (2) (2022) 634,
- [6]. S.S. Babu, A. Raha, M.K. Naskar, Trust evaluation based on node's characteristics and neighbouring nodes' recommendations for WSN, *Wirel. Sens. Netw.* 06 (08) (2014) 157-172,
- [7]. Raza, S., Wallgren, L., & Voigt, T. (2013). "Security and performance analysis of RPL for low-power and lossy

- networks." *Journal of Network and Computer Applications*, 37, 301-312.
- [8]. Mayzaud, A., Badonnel, R., & Chrisment, I. (2016). "A taxonomy of attacks in RPL-based Internet of Things." *International Journal of Communication Systems*, 29(12), e2398.
- [9]. Tripathi, S., Lal, C., Conti, M., & Jurdak, R. (2020). "Lightweight IDS for RPL-based IoT networks." *IEEE Transactions on Information Forensics and Security*, 15, 1880-1892.
- [10]. Sicari, S., Rizzardi, A., Coen-Porisini, A., & Cappiello, C. (2015). "A trust-based security framework for IoT." *Future Generation Computer Systems*, 45, 464-474.
- [11]. Shabut, A. M., Dahal, K., & Bista, S. (2018). "Trust management in IoT: A distributed approach." *Computer Networks*, 139, 43-57.
- [12]. Jabeur, N., El Falou, A., & Pierre, S. (2019). "Trust-based anomaly detection using Random Forest for secure IoT routing." *Wireless Communications and Mobile Computing*, 2019, 1-11.
- [13]. Wang, H., Li, X., & Jiang, Y. (2022). "XGBoost for anomaly detection in IoT networks." *IEEE Access*, 10, 6723-6735.
- [14]. Zhang, Y., Wang, J., & Chen, X. (2021). "DBN-TSP: A deep belief network-based trust model for IoT security." *Sensors*, 21(4), 987.
- [15]. Ali, S., Khan, N., & Lee, Y. (2023). "Hybrid CNN-LSTM for anomaly detection in IoT networks." *IEEE Transactions on Neural Networks and Learning Systems*, 34(5), 2234-2248.
- [16]. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*, 2016, pp. 785-794.
- [17]. Y. Tianqi, "Introduction to Boosted Trees and XGBoost," *arXiv preprint arXiv:1603.02754*, 2016.
- [18]. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Back propagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541-551, 1989.
- [19]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097-1105.
- [20]. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [21]. A. Graves, A.-R. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6645-6649.
- [22]. L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.
- [23]. A. Liaw and M. Wiener, "Classification and Regression by RandomForest," *R News*, vol. 2, no. 3, pp. 18-22, 2002.
- [24]. G. E. Hinton, S. Osindero, and Y. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, pp. 1527-1554, 2006.
- [25]. R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann Machines," in *Proc. 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009, pp. 448-455.