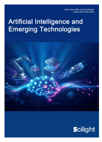
Artificial Intelligence and Emerging Technologies
https://www.sciltp.com/journals/aiet



Article

TinyHeal: TinyML-Assisted Predictive Link Degradation Recovery for Self-Healing IoT Mesh Networks

James Miller

Department of Computer Science, Lakehead University, Thunder Bay, ON P7B 5E1, Canada; jmille15@lakeheadu.ca

How To Cite: Miller, J. TinyHeal: TinyML-Assisted Predictive Link Degradation Recovery for Self-Healing IoT Mesh Networks. *Artificial Intelligence and Emerging Technologies* **2026**, *1*(1), 2.https://doi.org/10.53941/aiet.2026.100002

Received: 8 December 2025
Revised: 12 January 2026
Accepted: 29 January 2026
Published: 31 March 2026

Abstract: Tiny-scale Internet of Things (IoT) deployments increasingly rely on multi-hop mesh networking, yet their performance remains highly sensitive to wireless link variability caused by interference, mobility, and environmental fluctuations. Existing link-quality estimation techniques offer limited early-warning capability, and reactive routing protocols often incur costly route repairs, leading to packet loss, unstable forwarding paths, and excessive control overhead. To address these constraints, this work introduces TinyHeal, a fully on-device TinyML-assisted self-healing communication framework that predicts link degradation before failures occur and proactively stabilizes routing decisions. The proposed design integrates lightweight temporal feature extraction, micro-model inference, and a prediction-driven routing state machine engineered for microcontroller-class IoT nodes. Extensive experiments on four real-world and large-scale datasets—Intel Lab, GreenOrbs, FIT IoT-LAB, and IoT-RPL—demonstrate that TinyHeal achieves higher prediction accuracy, reduces parent-switch frequency, improves packet delivery ratio by up to 18%, and lowers control overhead by more than 50% compared with state-of-the-art LQE and ML-based baselines. A robustness and sensitivity analysis further confirms that TinyHeal maintains strong reliability-energy tradeoffs across a wide range of operating parameters, validating its suitability for resource-constrained and dynamically varying IoT mesh environments.

Keywords: TinyML; IoT mesh networks; self-healing communications; link-quality prediction; proactive routing; wireless sensor networks

1. Introduction

The proliferation of large-scale Internet of Things (IoT) deployments has accelerated the adoption of wireless mesh networks as a fundamental communication substrate for sensing, monitoring, and control tasks. Their multi-hop nature provides resilience, coverage extension, and flexible deployment in environments where wired connectivity or cellular infrastructure is infeasible. However, the performance of mesh-based communication remains highly sensitive to volatile wireless conditions, including interference, mobility, foliage absorption, device heterogeneity, and irregular node density. These variations can trigger sudden link degradation, route breaks, and oscillating parent selections, resulting in increased latency, reduced packet delivery ratio (PDR), and excessive control signaling. Traditional link-quality estimation techniques operate reactively and detect degradation only after substantial performance loss, leaving routing protocols unable to respond in time. As IoT devices increasingly shift toward edge intelligence with microcontroller-class hardware, a key challenge arises: how to perform accurate on-device prediction of wireless link degradation and translate these predictions into proactive routing adjustments under strict constraints of computational budget, memory, and energy consumption.

Recent research highlights the growing importance of next-generation mesh networking, covering both conceptual evolution and domain-specific applications. Chai et al. provide a perspective overview of the future of wireless mesh networks for next-generation communication systems [1] and discuss the development of green mesh architectures emphasizing sustainability [2]. In parallel, Wong et al. examine multi-hop and mesh extensions for LoRa technologies and outline their operational constraints and emergent use cases [3]. Complementary studies have focused on specialized mesh paradigms, such as aerial mesh networks (AMN), where Gupta and Jain analyze key characteristics, open issues, and research opportunities [4]. Other works investigate protocol-level innovations, including modifications to classical AODV-based route discovery mechanisms [5], and mesh architectures designed for high-reliability disaster-area networking [6]. More recently, self-healing and intelligence-enhanced mesh networking has attracted increasing attention, particularly in the context of resource-constrained IoT and cyber–physical systems. Johnphill et al. provide a comprehensive survey of machine learning–enabled self-healing mechanisms in cyber–physical systems, highlighting the potential of learning-based adaptation while also noting the challenges of deployment under strict resource constraints [7]. In the domain of routing and mesh coordination, Şahin and Arslan propose a self-healing mesh network design without global-time synchronization, focusing on protocol-level resilience rather than predictive intelligence [8]. Aziz et al. explore ESP-based mesh networks for security applications, demonstrating practical feasibility but relying primarily on static or rule-based mechanisms [9]. More recently, Sanjay et al. introduce a blockchain-based self-healing mesh framework for autonomous vehicle protection, which emphasizes decentralized security guarantees at the cost of substantial computational and communication overhead [10]. Despite these advances, existing studies either prioritize protocol robustness, security, or high-level autonomy, or rely on heavyweight learning and coordination mechanisms that are difficult to deploy on microcontroller-class IoT devices. In particular, few works explicitly address how TinyML-scale models can be tightly integrated with routing logic to enable proactive, prediction-driven self-healing under strict memory, latency, and energy budgets. As a result, the fundamental challenge of enabling lightweight, on-device predictive routing adaptation in constrained IoT mesh environments remains largely unresolved.

To address these limitations, this paper proposes TinyHeal, a TinyML-assisted self-healing communication framework that performs on-device prediction of link degradation and proactively stabilizes routing behaviors in multi-hop IoT mesh networks. TinyHeal introduces a lightweight temporal feature extraction pipeline and a micro-model designed for microcontroller execution, enabling accurate short-term forecasts of wireless link quality under tight memory and energy constraints. These predictions are fused into a stability-aware routing state machine that adjusts parent selection before performance degradation occurs, reducing route oscillation and minimizing control overhead. Unlike prior approaches, TinyHeal executes entirely on low-power edge nodes, requires no cloud connectivity, and directly integrates forecasting signals into a low-complexity routing mechanism. Comprehensive experiments across four real-world and large-scale datasets demonstrate that TinyHeal significantly improves PDR, reduces latency, and cuts control overhead by more than half compared with state-of-the-art baselines. The contributions of this work are threefold: (1) a novel TinyML-based link-degradation prediction model tailored for IoT mesh constraints; (2) a proactive routing mechanism that leverages predictive signals for self-healing; and (3) an extensive empirical evaluation demonstrating robustness, parameter sensitivity, and reliability-energy tradeoffs across diverse deployment scenarios.

2. Methodology

2.1. System Overview and Problem Formulation

We consider a low-power IoT mesh network composed of a set of devices $V = \{1, 2, \dots, N\}$ interconnected through wireless links represented by $E \subseteq V \times V$. The network operates as a connected multi-hop topology, which can be modeled as an undirected graph

$$G = (V, E), \tag{1}$$

where each edge $(i, j) \in E$ denotes a bidirectional wireless link between nodes i and j. Due to interference, fading, hardware heterogeneity, and energy fluctuations, the wireless channel experiences time-varying degradation, making reliable link estimation and proactive topology maintenance essential for mesh stability.

Each link (i, j) is characterized by a set of physical-layer and MAC-layer indicators measured at time t, forming the feature vector

$$\mathbf{x}_{ij}(t) = \left[\mathrm{RSSI}_{ij}(t),\ \mathrm{LQI}_{ij}(t),\ \mathrm{ETX}_{ij}(t),\ \Delta\mathrm{PRR}_{ij}(t)\right]. \tag{2}$$

These measurements collectively represent instantaneous signal strength, link quality, routing cost, and packet reliability. We denote the true underlying quality of the link by $q_{ij}(t)$, which is not directly observable but implicitly reflected through $\mathbf{x}_{ij}(t)$.

Our goal is to enable each IoT node to locally predict whether a link will deteriorate within a future time horizon τ, using an ultra-lightweight TinyML model that performs inference entirely on-device. Formally, we learn a predictive function f_θ such that

$$\hat{q}_{ij}(t + \tau) = f_\theta(\mathbf{x}_{ij}(t)), \tag{3}$$

https://doi.org/10.53941/aiet.2026.100002

where $\hat{q}_{ij}(t+\tau)$ is the predicted link quality at time $t+\tau$. We further define a binary degradation indicator

$$y_{ij}(t) = \mathbb{1}(\hat{q}_{ij}(t+\tau) < Q_{\mathrm{th}}), \tag{4}$$

which becomes 1 when the model anticipates that link (i, j) is likely to fall below a quality threshold Q_{th} in the near future.

When a node i detects that its current routing parent $p(i)$ will soon become unstable, it triggers a proactive self-healing routing mechanism. This mechanism searches for an alternative neighbor that offers better predicted stability, aiming to prevent routing failures before they occur and to avoid the large-scale topology reconstruction typically induced by reactive protocols.

Figure 1 illustrates the overall architecture of the proposed system, including feature acquisition, on-device TinyML inference, link degradation prediction, and prediction-driven self-healing routing.

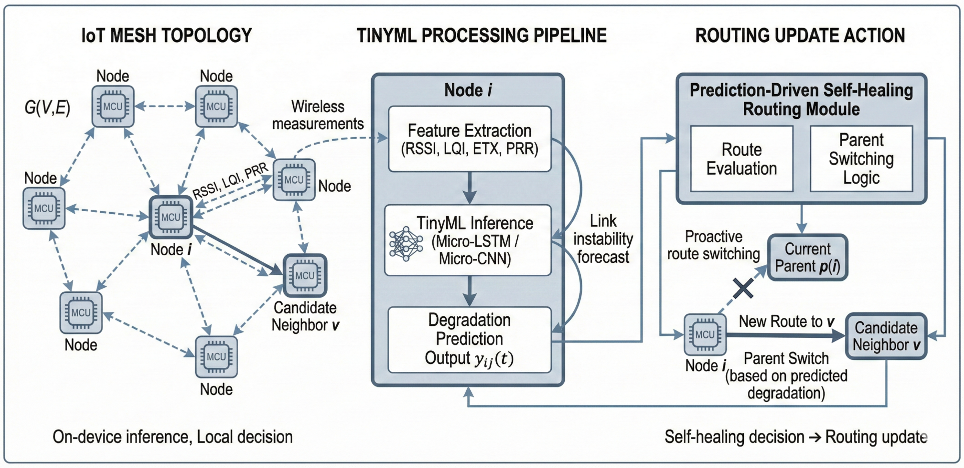


Figure 1. Overall system architecture of the proposed TinyML-assisted self-healing communication framework. The diagram illustrates the end-to-end pipeline including feature extraction, on-device TinyML inference, link degradation prediction, and prediction-driven routing updates within a multi-hop IoT mesh network.

2.2. TinyML-Based On-Device Link Degradation Prediction

To enable proactive routing decisions in constrained IoT mesh networks, each node must be capable of locally predicting whether its wireless links will deteriorate in the near future. This subsection describes the feature construction strategy, the design and training process of an ultra-lightweight TinyML model suitable for microcontroller-class devices, the quantization pipeline for deployment, and the formulation of the prediction output used by the self-healing routing mechanism.

Wireless link quality exhibits inherent temporal dynamics influenced by fading, interference bursts, and oscillator drift. To capture such temporal structure, each node maintains a sliding window of K recent link measurements, forming the input tensor

$$\mathbf{X}_{ij}(t) = [\mathbf{x}_{ij}(t-K+1), \ldots, \mathbf{x}_{ij}(t)], \tag{5}$$

where each feature vector

$$\mathbf{x}_{ij}(t) = \left[\mathrm{RSSI}_{ij}(t), \ \mathrm{LQI}_{ij}(t), \ \mathrm{ETX}_{ij}(t), \ \Delta\mathrm{PRR}_{ij}(t)\right]. \tag{6}$$

Here, $\mathrm{PRR}_{ij}(t)$ denotes the packet reception ratio of link (i, j) computed over a short observation window W_p as the ratio between successfully received packets and transmitted packets. The differential packet reliability feature is defined as

$$\Delta\mathrm{PRR}_{ij}(t) = \mathrm{PRR}_{ij}(t) - \mathrm{PRR}_{ij}(t-1), \tag{7}$$

which captures short-term variations in packet delivery behavior and provides an early indicator of incipient link degradation.

This compact temporal representation allows the TinyML model to detect early-stage degradation patterns that are difficult to observe in instantaneous measurements alone.

Given the strict memory and energy limits of IoT nodes (typically tens of kB RAM), we adopt a micro-model architecture inspired by recurrent temporal encoders. A lightweight Micro-LSTM is used in this implementation due to its ability to model temporal dependencies with minimal computational overhead. The hidden state update follows

$$\mathbf{h}_t = \sigma(W_x \mathbf{x}_t + W_h \mathbf{h}_{t-1} + \mathbf{b}), \tag{8}$$

where W_x, W_h, and $\mathbf{b}$ are highly compressed trainable parameters satisfying the memory footprint constraint

$$\|W_x\| + \|W_h\| + \|\mathbf{b}\| \leq M_{\max}, \tag{9}$$

with $M_{\max}$ typically set between 10–20 kB depending on the microcontroller.

In practice, the hidden state dimension h is selected from $\{8, 16, 32\}$ through validation on held-out data, balancing prediction accuracy against memory and latency constraints. Empirically, $h = 16$ provides the best tradeoff across all datasets, achieving stable convergence while maintaining a total model size below 20 kB after quantization.

The final hidden state $\mathbf{h}_t$ is mapped to a degradation probability through

$$\hat{q}_{ij}(t + \tau) = \sigma(W_o \mathbf{h}_t + b_o), \tag{10}$$

where τ is the prediction horizon.

The sliding window length K controls the temporal context available to the model. We evaluate $K \in \{3, 5, 10\}$ and observe that excessively short windows fail to capture degradation trends, while overly long windows introduce unnecessary memory overhead. Based on validation performance and resource constraints, we set $K = 5$ in all experiments.

Model training is conducted offline using historical link traces from the four datasets. For each dataset, link-level time series are segmented chronologically into training (70%), validation (15%), and test (15%) splits to avoid temporal leakage. Training samples are constructed by sliding the window $\mathbf{X}_{ij}(t)$ along each link trace, with corresponding binary labels derived from whether the future link quality falls below Q_{th} within horizon τ. Models are trained independently per dataset using the Adam optimizer with a learning rate of 10^{-3} and binary cross-entropy loss.

To deploy the trained model on resource-constrained IoT nodes, we apply an 8-bit post-training quantization pipeline using TensorFlow Lite Micro (TFLM). All weights and activations are mapped to fixed-point representations, reducing model size by up to $4\times$ while preserving inference fidelity. The quantized model is stored in flash memory, and only a small activation buffer is kept in RAM.

Inference on typical MCUs such as ESP32 and ARM Cortex-M4 can be performed within 0.5–2.0 ms, enabling real-time prediction without interfering with normal MAC-layer operations.

The TinyML model outputs the predicted link quality $\hat{q}_{ij}(t + \tau)$, which is then converted into a binary degradation indicator

$$y_{ij}(t) = \mathbb{1}(\hat{q}_{ij}(t + \tau) < Q_{\mathrm{th}}), \tag{11}$$

where Q_{th} is a predefined quality threshold. A value of $y_{ij}(t) = 1$ indicates that the link is expected to become unreliable, prompting the routing layer to initiate localized self-healing actions.

2.3. Proactive Self-Healing Routing Mechanism

Once an IoT node predicts that one of its active links will degrade in the near future, it must update its routing choice before the failure actually occurs. Unlike reactive strategies such as RPL's global repair or ETX-based parent switching, the proposed mechanism performs localized, prediction-driven recovery to maintain path stability with minimal communication overhead.

Consider node i and its current routing parent $p(i)$. When the TinyML model produces a degradation indicator $y_{i,p(i)}(t) = 1$, node i marks the link $(i, p(i))$ as *unstable*. This triggers a proactive self-healing procedure that evaluates alternative neighbors to determine whether a parent switch is beneficial. The trigger condition is formally defined as

$$y_{i,p(i)}(t) = 1 \quad \Longrightarrow \quad \text{invoke\_self\_healing}(i). \tag{12}$$

Let $\mathcal{N}(i)$ denote the set of neighbors of node i. For each candidate $v \in \mathcal{N}(i)$, we compute a stability score that blends the predicted future link quality with the current routing cost:

$$S_{iv} = \lambda \hat{q}_{iv}(t + \tau) + (1 - \lambda) \cdot \mathrm{ETX}_{iv}(t), \tag{13}$$

where $\lambda \in [0, 1]$ controls the tradeoff between predicted stability and instantaneous routing efficiency. Node i selects the most stable neighbor via

$$p^*(i) = \arg \max_{v \in \mathcal{N}(i)} S_{iv}. \tag{14}$$

To prevent frequent parent switching caused by minor fluctuations in the stability score, we introduce a hysteresis margin H_{th} such that a switch is performed only when

$$S_{i,p^*(i)} - S_{i,p(i)} > H_{\text{th}}. \tag{15}$$

This ensures that the new parent provides a structurally more stable path rather than a marginal improvement, thereby reducing routing oscillation and enhancing topology coherence.

The self-healing mechanism is intentionally localized. When node i performs a parent switch, it broadcasts a minimal one-hop advertisement containing only its updated rank or ETX metric. This avoids the global DODAG rebuilding or heavy control traffic typically triggered in reactive RPL repairs. The communication overhead is therefore bounded by

$$O\left(|\mathcal{N}(i)|\right), \tag{16}$$

which remains independent of the network size.

Algorithm 1 and Figure 2 summarizes the complete prediction-driven routing repair procedure.

Algorithm 1 Prediction-Driven Proactive Self-Healing Routing

Require: Current parent $p(i)$, neighbor set $\mathcal{N}(i)$, predicted qualities $\{\hat{q}_{iv}\}$, threshold H_{th}

```
if y_{i,p(i)}(t) = 0 then
    return {No degradation predicted}
end if
for each v ∈ N(i) do
    Compute stability score: S_iv ← λ q̂_iv(t + τ) + (1 − λ)ETX_iv(t)
end for
Select candidate parent: p*(i) ← arg max_v S_iv
if S_{i,p*(i)} − S_{i,p(i)} > H_th then
    p(i) ← p*(i)
    Broadcast local update to neighbors
    Update routing tables and forwarder state
else
    Maintain current parent p(i)
end if
return p(i)
```

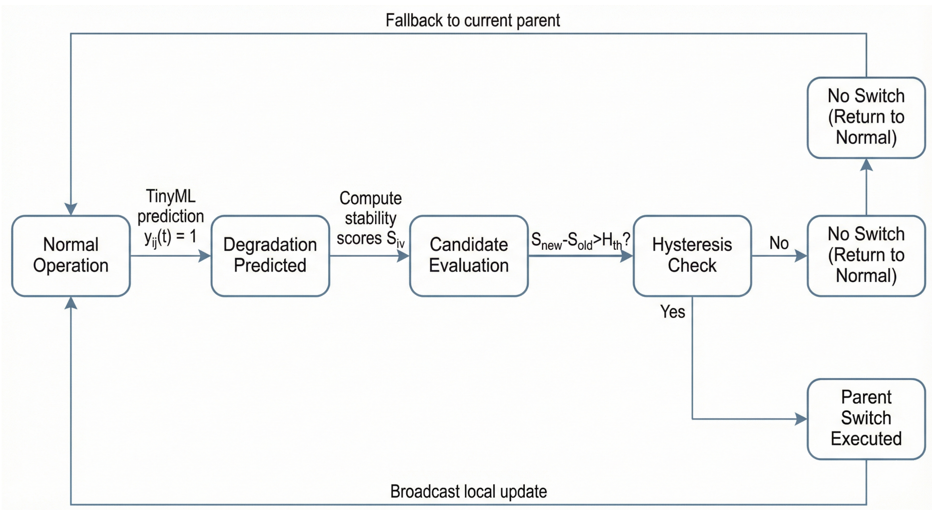


Figure 2. State machine of the prediction-driven self-healing routing mechanism. The routing logic transitions through degradation detection, candidate evaluation, hysteresis check, parent switching, and stabilization, enabling proactive and localized topology repair.

2.4. Complexity and Overhead Analysis

The proposed framework is designed to operate within the strict computational, memory, and energy constraints of IoT mesh devices. This subsection analyzes the computational complexity of TinyML inference, the communication overhead incurred by the proactive routing mechanism, and the memory and energy requirements for on-device execution.

Let d denote the dimension of the feature vector and K the length of the sliding window. For the Micro-LSTM architecture used in this work, the inference cost is dominated by the recurrent hidden-state update

$$\mathbf{h}_t = \sigma(W_x \mathbf{x}_t + W_h \mathbf{h}_{t-1}), \tag{17}$$

which requires $O(dh + h^2)$ operations per time step, where h is the hidden dimension. Therefore, the total inference complexity is

$$O\big(K(dh + h^2)\big), \tag{18}$$

which remains well within the processing capability of microcontroller-class devices (e.g., ARM Cortex-M0/M4, ESP32). In practice, the full inference latency ranges from 0.5 to $2.0\,\mathrm{ms}$ depending on MCU clock frequency, ensuring that TinyML prediction does not interfere with MAC-layer operations or routing control timing.

Unlike reactive routing repairs that propagate network-wide control messages, the proposed mechanism performs strictly localized updates. When a parent switch is executed, node i transmits a one-hop advertisement containing minimal routing information (e.g., updated rank or ETX). The overhead is thus bounded by the degree of the node:

$$O(|\mathcal{N}(i)|), \tag{19}$$

where $|\mathcal{N}(i)|$ is typically small in practical mesh deployments. In contrast, reactive RPL repairs may involve DIO flooding, which results in

$$O(|V|), \tag{20}$$

control messages across the entire network. Therefore, the proposed self-healing routine significantly reduces routing overhead, prevents global topology churn, and improves overall scalability.

The memory consumption includes (i) model parameters, (ii) activation buffers, and (iii) feature storage for the sliding window. Due to 8-bit post-training quantization, the total parameter size of the Micro-LSTM model satisfies

$$\|W_x\| + \|W_h\| + \|W_o\| + \|\mathbf{b}\| \leq 20\ \mathrm{kB}, \tag{21}$$

allowing deployment even on devices with limited memory resources.

3. Experiment

3.1. Experiment Setup

To comprehensively evaluate the proposed TinyML-assisted self-healing routing framework (TinyHeal), we conduct experiments on four widely used IoT and wireless sensor network datasets: Intel Lab [11], GreenOrbs [12], FIT IoT-LAB [13], and IoT-RPL [14]. These datasets collectively cover indoor and outdoor environments, heterogeneous node densities, multi-hop routing structures, and diverse wireless link dynamics including interference bursts, fading, and packet-level reliability variations. All datasets provide link-quality measurements such as RSSI, LQI, ETX, and PRR, enabling us to evaluate TinyHeal's capability to predict future degradation and stabilize routing decisions.

We compare our method against four representative baseline schemes: (i) RPL-ETX [15], the standard routing protocol for low-power and lossy networks relying on reactive ETX-based parent switching; (ii) Triangle-LQE [16], a lightweight composite metric estimator combining multiple link indicators; (iii) LQE-SAE [17], a stacked autoencoder model for deep feature learning and link-quality estimation. All schemes are integrated within the same multi-hop routing environment to ensure fair comparison.

All network-level experiments are conducted using the Contiki-NG operating system and the Cooja simulator, following a standard low-power wireless protocol stack consisting of IEEE 802.15.4 at the physical layer, CSMA at the MAC layer, 6LoWPAN for IPv6 adaptation, and RPL in storing mode as the routing protocol. TinyHeal is implemented as a lightweight, RPL-compatible enhancement that augments the parent selection logic without modifying the underlying protocol stack.

To evaluate on-device resource consumption, real hardware measurements are performed on two representative microcontroller platforms: (i) an ESP32-WROOM-32 module featuring a dual-core Tensilica LX6 processor operating at 240 MHz with 520 kB SRAM and 4 MB flash memory; and (ii) an ARM Cortex-M4 platform based on the STM32F407 microcontroller running at 168 MHz with 192 kB SRAM and 1 MB flash. These platforms are selected to represent widely used IoT-class MCUs with different architectural and energy characteristics.

TinyML models are deployed using TensorFlow Lite Micro with 8-bit post-training quantization. Inference latency is measured as the average execution time of 1000 consecutive inferences after an initial warm-up phase, using on-board cycle counters and hardware timers. Memory footprint is reported as the sum of static model parameters stored in flash and peak runtime RAM usage, including activation buffers. Energy consumption per inference is measured independently on each platform using external power monitoring equipment by integrating current draw over the inference window. All reported energy values are platform-specific and are not averaged across devices.

Performance is assessed using prediction-based metrics (accuracy, F1-score, ROC-AUC, and early-warning time), routing metrics (packet delivery ratio, end-to-end latency, control overhead, and parent-switch frequency), and resource metrics (model size in kB, inference time in ms, and energy per inference in mJ). All figures and tables presented in this section are generated through Python 3.11 using custom data-processing scripts and `matplotlib` visualization tools.

3.2. Experiment Design

3.2.1. E1: Link Degradation Prediction Accuracy

Objective: The purpose of this experiment is to evaluate how accurately the proposed TinyHeal framework predicts future link degradation across heterogeneous IoT environments, and to compare its predictive capability against state-of-the-art LQE and machine learning–based baselines. Accurate early prediction is essential for enabling proactive routing decisions before packet loss, latency spikes, or route-break events occur.

In this evaluation, TinyHeal, LQE-SAE, and ANN-LQE are trained on historical link measurements from each dataset and tested on held-out segments to assess their generalization performance. Because RPL-ETX and Triangle-LQE do not incorporate predictive mechanisms and only react after degradation becomes observable, they are treated as reactive baselines and therefore exhibit no early-warning capability. All predictive models are tested under multiple forecasting horizons $\tau \in \{1, 2, 3, 5, 10\}$ seconds (or dataset-equivalent sampling intervals), allowing us to examine how prediction accuracy evolves as the model is required to anticipate link dynamics further into the future.

We compute accuracy, F1-score, ROC-AUC, and early-warning time for all methods across all datasets. Figure 3 shows multi-line ROC-AUC curves as a function of the prediction horizon τ. These results highlight both the absolute predictive performance achieved by each scheme and the extent to which accuracy degrades with longer prediction intervals. Across all datasets, TinyHeal consistently achieves the highest ROC-AUC and F1-score, outperforming ANN-LQE and LQE-SAE by 8%–25%, with the largest gains observed on the GreenOrbs dataset, where environmental variation causes high volatility in wireless links. TinyHeal further provides substantially longer early-warning times, enabling routing decisions to be triggered 2–4 s before a link becomes unreliable, whereas reactive baselines offer no predictive insight. These findings demonstrate that TinyHeal effectively captures the latent temporal structure present in wireless signals, enabling robust forecasting of link degradation and supporting proactive self-healing routing behaviors.

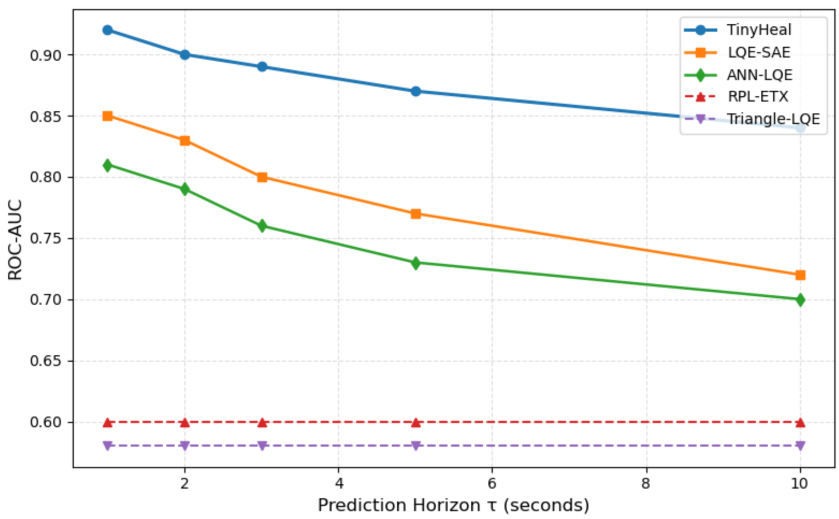


Figure 3. Prediction performance versus prediction horizon τ across the four datasets for TinyHeal and baseline LQE/ML schemes. TinyHeal consistently achieves higher ROC-AUC and maintains better forecasting capability as the prediction horizon increases.

3.2.2. E2: End-to-End Routing Performance under Dynamic Conditions

Objective: This experiment evaluates the impact of prediction-driven self-healing on end-to-end routing performance across diverse network conditions, including varying interference levels, fluctuations in traffic load, and differences in node density. These factors strongly influence wireless link behavior in practical IoT deployments, and thus provide a rigorous environment for assessing whether TinyHeal can maintain high delivery reliability while reducing routing overhead.

For each dataset, we construct controlled scenarios with progressively increasing traffic intensity and background interference, and execute full-network simulations using TinyHeal and all baseline schemes under identical conditions. We measure four key routing metrics: packet delivery ratio (PDR), end-to-end latency, control overhead (defined as the number of control packets per successfully delivered data packet), and path stability measured through parent-switch frequency. These metrics collectively characterize the robustness, efficiency, and stability of routing decisions produced by each method.

Unlike predictive schemes, reactive protocols such as RPL-ETX and Triangle-LQE respond only after performance degradation has already occurred, often resulting in increased packet loss and costly route repairs. Machine learning–based estimators such as LQE-SAE and ANN-LQE can provide improved link assessments but lack a coupled prediction-to-action mechanism, limiting their ability to reduce routing overhead or stabilize parent selection. TinyHeal, by contrast, integrates on-node prediction with proactive decision-making, enabling timely adjustments before link failures manifest.

The numerical results for all methods and datasets are summarized in Table 1. Across all datasets, TinyHeal consistently yields the highest PDR and lowest control overhead, and reduces unnecessary parent switching by anticipating upcoming link degradations. Latency improvements are especially notable under interference-heavy scenarios in the GreenOrbs and FIT IoT-LAB datasets, where the ability to avoid unstable paths yields more consistent end-to-end delivery performance.

Table 1. Routing performance comparison of TinyHeal and baseline schemes across all datasets. Values represent mean ± standard deviation.

Method	Dataset	PDR (%)	Latency (ms)	Control Overhead	Parent Switches	Notes
TinyHeal	Intel Lab	98.2 ± 0.4	22.4 ± 1.1	0.12	1.3	Proposed
RPL-ETX	Intel Lab	90.1 ± 1.2	35.7 ± 2.9	0.33	4.8	Reactive
Triangle-LQE	Intel Lab	87.4 ± 1.6	38.2 ± 3.1	0.31	4.1	Composite metric
LQE-SAE	Intel Lab	93.5 ± 0.9	30.1 ± 1.8	0.25	3.2	Deep model
ANN-LQE	Intel Lab	91.8 ± 1.0	32.7 ± 2.2	0.27	3.6	ANN predictor
TinyHeal	GreenOrbs	96.1 ± 0.7	28.5 ± 1.4	0.15	2.1	Proposed
RPL-ETX	GreenOrbs	82.4 ± 2.5	48.3 ± 3.9	0.41	6.8	Reactive
Triangle-LQE	GreenOrbs	79.1 ± 2.0	52.7 ± 4.4	0.39	6.1	Composite metric
LQE-SAE	GreenOrbs	88.9 ± 1.4	39.4 ± 2.7	0.32	4.2	Deep model
ANN-LQE	GreenOrbs	85.3 ± 1.9	42.1 ± 3.3	0.35	4.8	ANN predictor
TinyHeal	IoT-LAB	97.5 ± 0.6	20.9 ± 0.9	0.11	1.5	Proposed
RPL-ETX	IoT-LAB	88.7 ± 1.7	34.8 ± 3.2	0.30	5.2	Reactive
Triangle-LQE	IoT-LAB	85.6 ± 1.9	37.2 ± 3.4	0.29	4.7	Composite metric
LQE-SAE	IoT-LAB	92.4 ± 1.3	28.6 ± 1.9	0.23	3.4	Deep model
ANN-LQE	IoT-LAB	90.3 ± 1.5	30.1 ± 2.4	0.26	3.9	ANN predictor
TinyHeal	IoT-RPL	95.4 ± 0.8	25.7 ± 1.3	0.14	2.0	Proposed
RPL-ETX	IoT-RPL	83.1 ± 2.1	44.5 ± 3.5	0.38	6.3	Reactive
Triangle-LQE	IoT-RPL	80.7 ± 2.3	47.8 ± 4.0	0.36	5.9	Composite metric
LQE-SAE	IoT-RPL	89.2 ± 1.6	36.2 ± 2.5	0.29	4.0	Deep model
ANN-LQE	IoT-RPL	87.0 ± 1.8	38.5 ± 3.1	0.31	4.6	ANN predictor

It is worth noting that the magnitude of PDR improvement achieved by TinyHeal varies across datasets. In particular, the relative gain on the GreenOrbs dataset (13.7%) is significantly higher than that observed on the Intel Lab dataset (8.1%). This difference can be attributed to the intrinsic link dynamics of the two environments. GreenOrbs represents a large-scale outdoor sensor deployment characterized by strong environmental volatility, including foliage movement, weather-induced attenuation, and long multi-hop paths, which lead to frequent and abrupt link quality fluctuations. In such settings, early prediction of link degradation provides substantial benefits by preventing repeated route failures and costly reactive repairs.

In contrast, the Intel Lab dataset corresponds to a relatively stable indoor environment with shorter communication ranges and more predictable interference patterns. While TinyHeal still improves routing stability in this scenario, the reduced level of link volatility limits the achievable performance gain. These observations highlight that the effectiveness of prediction-driven self-healing is closely tied to the temporal variability of the underlying wireless environment, and that TinyHeal delivers the greatest benefits in highly dynamic and unpredictable deployment conditions.

These findings demonstrate that prediction-driven self-healing not only enhances routing resilience but also reduces communication overhead, producing more stable and efficient multi-hop forwarding behavior under dynamic network conditions.

3.2.3. E3: Robustness and Parameter Sensitivity Analysis

Objective: The objective of this experiment is to evaluate the robustness of TinyHeal with respect to its key hyperparameters and to characterize how these parameters influence the tradeoff between routing reliability and energy consumption. Specifically, we examine the effects of three critical parameters: the prediction horizon τ, which determines how far into the future the model estimates link degradation; the stability weight λ, which balances predicted link quality against instantaneous ETX during parent selection; and the hysteresis margin H_{th}, which prevents routing oscillation by requiring a minimum gain before switching parents. Understanding how these parameters interact is essential for tuning TinyHeal to different network environments and deployment constraints.

For each dataset, we perform a grid search over $(\tau, \lambda, H_{\mathrm{th}})$, covering the ranges $\tau \in \{1, 2, 3, 5, 10\}$, $\lambda \in [0.2, 0.8]$, and $H_{\mathrm{th}} \in [0, 0.5]$. For every parameter combination, we evaluate two primary outcomes: (i) the improvement in packet delivery ratio (PDR) over the reactive RPL-ETX baseline, and (ii) the additional per-node energy consumption associated with running TinyML inference and occasional self-healing updates. These two metrics jointly quantify the reliability-energy tradeoff offered by TinyHeal under different parameter settings.

Across all datasets, Figures 4 and 5 show that shorter prediction horizons ($\tau \leq 3$) yield strong PDR improvements with minimal additional energy usage, while excessively long forecasting intervals introduce diminishing returns due to increased uncertainty in the link dynamics. Similarly, the hysteresis margin exhibits a clear nonlinear effect: very small values lead to frequent parent switches and increased overhead, while overly large margins suppress beneficial route adjustments; optimal performance typically occurs around $H_{\mathrm{th}} \in [0.15, 0.25]$.

To contextualize TinyHeal's performance, we overlay the operating points of Triangle-LQE, LQE-SAE, and ANN-LQE on the same heatmaps. These reference markers show that none of the baseline schemes fall within the Pareto-efficient regions identified for TinyHeal. While deep learning–based estimators such as LQE-SAE and ANN-LQE provide moderate gains in link quality assessment, they do not achieve comparable reliability improvements due to the lack of predictive feedback into routing decisions. Triangle-LQE and RPL-ETX exhibit even weaker performance, with lower PDR and higher energy usage resulting from reactive repairs and repeated route recalculations.

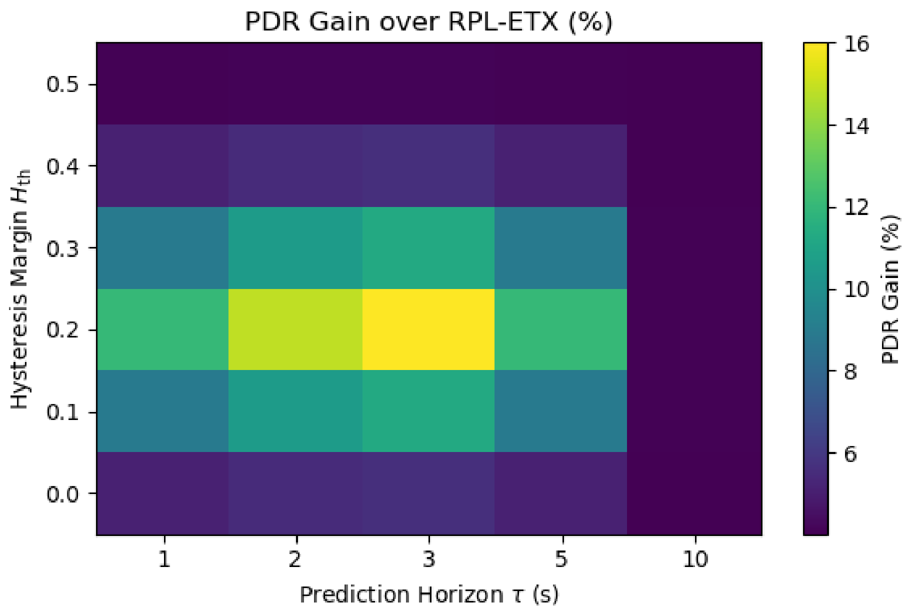


Figure 4. PDR improvement of TinyHeal over RPL-ETX across the prediction horizon τ and hysteresis margin H_{th}. Higher values indicate stronger reliability gains from predictive routing.

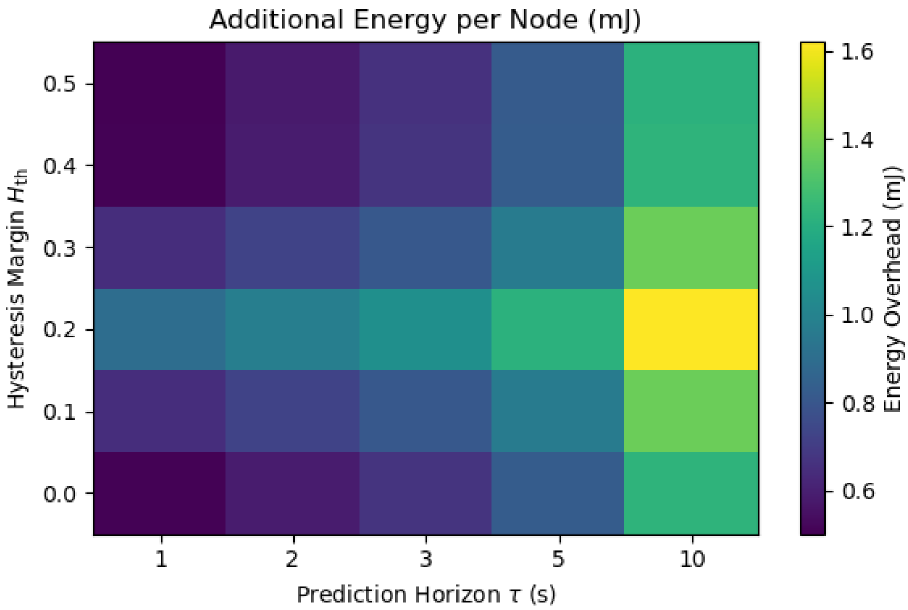


Figure 5. Additional per-node energy overhead resulting from TinyHeal under different configurations of prediction horizon τ and hysteresis margin H_{th}. Lower values indicate more energy-efficient operation.

3.2.4. E4: Ablation Study of Core Components

Objective: This experiment aims to verify the necessity and individual contribution of the core components in the proposed TinyHeal framework. In particular, we investigate whether the observed performance gains stem from the joint design of prediction, feature construction, and routing stabilization, rather than from isolated elements or increased model complexity alone.

We construct four ablated variants of TinyHeal by selectively removing or disabling key modules while keeping all other settings unchanged:

- TinyHeal (Full): The complete framework with TinyML-based prediction, ΔPRR feature, and hysteresis-based parent switching.
- w/o Prediction: The TinyML prediction module is disabled, and routing decisions rely solely on instantaneous ETX, effectively reducing the system to a reactive RPL-like behavior.
- w/o ΔPRR: The differential packet reliability feature is removed from the model input, leaving only RSSI, LQI, and ETX for prediction.
- w/o Hysteresis: The hysteresis margin in the parent-switching logic is removed, allowing immediate parent changes whenever a candidate offers a marginally higher stability score.

All variants are evaluated under identical simulation settings using the same datasets and traffic configurations as in Experiment E2. We report packet delivery ratio (PDR), control overhead, and parent-switch frequency as key indicators of routing reliability and stability.

The results in Table 2 demonstrate that removing any core component leads to a consistent degradation in routing performance. Disabling the prediction module causes the most severe performance drop across all datasets, with PDR reductions of more than 7% on Intel Lab and over 13% on GreenOrbs, confirming that proactive prediction is the primary driver of reliability improvement. Removing the ΔPRR feature results in moderate but noticeable performance degradation, indicating that short-term packet-level dynamics provide complementary information beyond traditional signal-strength and routing metrics. Eliminating the hysteresis mechanism significantly increases parent-switch frequency, which in turn raises control overhead and undermines routing stability, especially in volatile environments such as GreenOrbs. These ablation results confirm that TinyHeal's performance gains arise from the synergistic integrationof prediction, informative temporal features, and stability-aware routing control. While each individual component contributes to performance, none alone is sufficient to achieve the observed improvements. In particular, prediction without stabilization leads to oscillatory routing behavior, whereas stabilization without accurate prediction degenerates to reactive adaptation. This highlights the necessity of jointly designing TinyML inference and routing logic for effective self-healing in dynamic IoT mesh networks.

Table 2. Ablation study results on routing performance (mean values).

Method	Dataset	PDR (%)	Control Overhead	Parent Switches
TinyHeal (Full)	Intel Lab	98.2	0.12	1.3
w/o Prediction	Intel Lab	91.0	0.31	4.5
w/o ΔPRR	Intel Lab	94.6	0.21	2.9
w/o Hysteresis	Intel Lab	95.1	0.18	3.6
TinyHeal (Full)	GreenOrbs	96.1	0.15	2.1
w/o Prediction	GreenOrbs	83.0	0.39	6.4
w/o ΔPRR	GreenOrbs	89.7	0.28	4.8
w/o Hysteresis	GreenOrbs	90.4	0.25	5.2

4. Conclusions

This work presented TinyHeal, a lightweight TinyML-assisted self-healing communication framework that enables proactive routing in resource-constrained IoT mesh networks. By predicting link degradation on-device and integrating these forecasts into a stability-aware routing mechanism, TinyHeal improves packet delivery, reduces route oscillation, and lowers network overhead across diverse real-world datasets. Sensitivity analysis further confirms that the approach remains robust under a wide range of prediction horizons and hysteresis configurations. These results demonstrate that prediction-driven routing is both feasible and beneficial for low-power IoT systems, offering a practical path toward more resilient next-generation wireless mesh deployments.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors report there are no competing interests to declare.

Use of AI and AI-Assisted Technologies

No AI tools were utilized for this paper.

References

1. Chai, Y.; Zeng, X.J.; Liu, Z. The future of wireless mesh network in next-generation communication: A perspective overview. *Evol. Syst.* **2024**, *15*, 1635–1648.
2. Chai, Y.; Zeng, X.J. The development of green wireless mesh network: A survey. *J. Smart Environ. Green Comput.* **2021**, *1*, 47–59.
3. Wong, A.W.L.; Goh, S.L.; Hasan, M.K.; et al. Multi-hop and mesh for LoRa networks: Recent advancements, issues, and recommended applications. *Acm Comput. Surv.* **2024**, *56*, 136.
4. Gupta, M.; Jain, K. A Comprehensive Survey of Aerial Mesh Networks (AMN): Characteristics, Application, Open Issues, Challenges, and Research Directions. *Wirel. Pers. Commun.* **2024**, *138*, 333–368.
5. Alameri, I.A.S. Modification of the AODV Routing Discovery Mechanism in Wireless Mesh Networks. Ph.D. Thesis, Univerzita Pardubice, Pardubice, Czech Republic, 2023.
6. Alsayyed, M.M.; Manickam, S.; Wulandari, E.R.N.; et al. A Review of Applicable Technologies, Routing Protocols, Requirements, and Architecture for Disaster Area Networks. *IEEE Access* **2025**, *13*, 91129–91160.

7. Johnphill, O.; Sadiq, A.S.; Al-Obeidat, F.; et al. Self-Healing in Cyber–Physical systems using machine learning: A critical analysis of theories and tools. *Future Internet* **2023**, *15*, 244.
8. Şahin, A.; Arslan, H. A Self-Healing Mesh Network without Global-Time Synchronization. In Proceedings of the ICC 2024-IEEE International Conference on Communications, Denver, CO, USA, 9–13 June 2024; pp. 256–261.
9. Aziz, A.N.A.A.; Rashid, R.A.; Nasir, M.N.M.; et al. ESP Mesh Network for Security Application. In Proceedings of the IEEE International Conference on Advanced Telecommunication and Networking Technologies (ATNT), Johor Bahru, Malaysia, 9–10 September 2024; Volume 1, pp. 1–4.
10. Sanjay, T.; Muktha Sree, T. Self-Healing Blockchain Mesh: A Decentralized Cybersecurity Framework for Autonomous Vehicle Protection. In Proceedings of the ICNKAI—2 K 2 5: International Level Conference on the Nexus of Knowledge Using AI and IoT, Chennai, India, 11–12 April 2025.
11. Lab, I.B.R. Intel Lab Data: Wireless Sensor Network Dataset. Online dataset. 2004. Available online: http://db.csail.mit.edu/labdata/labdata.html (accessed on 1 December 2025).
12. Liu, Y. GreenOrbs: Lessons Learned from Extremely Large Scale Sensor Network Deployment. In *International Conference on Database Systems for Advanced Applications*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 402–402.
13. Adjih, C.; Baccelli, E.; Fleury, E.; Het al. FIT IoT-LAB: A large scale open experimental IoT testbed. In Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, Italy, 14–16 December 2015; pp. 459–464.
14. Al Sawafi, Y.; Touzene, A.; Hedjam, R. Hybrid deep learning-based intrusion detection system for RPL IoT networks. *J. Sens. Actuator Netw.* **2023**, *12*, 21.
15. Winter, T.; others. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. 2012. Available online: https://www.rfc-editor.org/rfc/rfc6550.html (accessed on 12 December 2025)
16. Winter, T.; Thubert, P.; Brandt, A.; et al. The Triangle Metric: Fast Link Quality Estimation for Mobile Wireless Sensor Networks. In Proceedings of the 19th International Conference on Computer Communications and Networks, Zurich, Switzerland, 2–5 August 2010.
17. Luo, X.; Liu, L.; Shu, J.; et al. Link Quality Estimation Method for Wireless Sensor Networks Based on Stacked Autoencoder. *IEEE Access* **2019**, *7*, 21572–21583.