Detecção de Anomalias e Falhas em Redes LoRa Mesh: Uma Abordagem com Isolation Forest e LLM para Redução de Falsos Positivos

Juliano Paulo de Freitas[1] e Flávio Henrique Teles Vieira[1]

[1]CERISE (Centro de Excelência em Redes Inteligentes e Serviços Avançados) - Escola de Engenharia Elétrica e Computação - Universidade Federal de Goiás (UFG)

Goiânia – GO - Brasil

juliano.paulo@discente.ufg.br, flavio_vieira@ufg.br

Abstract. *LoRa Mesh networks for the Internet of Things (IoT) face physical failures and topological instability. In these scenarios, wireless channel variability and benign route reconvergences increase false positives in monitoring, causing alert fatigue. This paper proposes a three-layer detection architecture: (i) unsupervised screening via Isolation Forest, (ii) deterministic rule-based classification, and (iii) semantic validation using a Large Language Model (LLM). Evaluated in a real testbed, the integration of the LLM as a contextual filter autonomously suppressed 89.1% of the false alerts generated by the baseline statistical model. The solution ensured the uninterrupted notification of critical anomalies through controlled degradation mechanisms.*

Resumo. *Redes LoRa Mesh para Internet das Coisas (IoT) enfrentam falhas físicas e instabilidade topológica. Nestes cenários, a variabilidade do canal sem fio e reconvergências benignas elevam os falsos positivos no monitoramento, causando fadiga de alertas. Este artigo propõe uma arquitetura de detecção composta por três camadas: (i) triagem não supervisionada via Isolation Forest, (ii) tipificação por regras determinísticas e (iii) validação semântica com Large Language Model (LLM). Avaliada em um testbed real, a integração do LLM como filtro contextual suprimiu autonomamente 89,1% dos falsos alertas originados pelo modelo estatístico. A solução garantiu a notificação ininterrupta de anomalias críticas por meio de mecanismos de degradação controlada.*

1. Introdução

Redes LoRa Mesh operam em condições que se intensificam em ambientes indoor: a atenuação por obstáculos físicos e o desvanecimento por multipercurso produzem condições de propagação altamente não lineares, fenômeno bem documentado na literatura de enlace LoRa [Solé et al. 2022]. Nesse cenário, abordagens de monitoramento baseadas em limiares fixos geram um volume excessivo de falsos positivos, levando à fadiga de alertas no operador. Técnicas não supervisionadas como o Isolation Forest [Liu et al. 2008] identificam desvios estatísticos sem necessidade de dados rotulados, mas não indicam a causalidade da falha. Eventos contextualmente esperados em redes mesh, como reconvergências de rota, flutuações de RSSI por reflexões do ambiente e instabilidade

topológica transitória, são sistematicamente classificados como anomalias por esses modelos.

Para superar essas limitações, este trabalho propõe uma arquitetura de detecção organizada em três camadas: (i) triagem estatística não supervisionada com Isolation Forest; (ii) tipificação determinística em sete classes operacionais (A1 a A7); e (iii) validação semântica por Large Language Model (LLM), que avalia o contexto temporal e topológico da malha para distinguir falhas reais de transientes inofensivos.

As principais contribuições deste artigo são: (i) a definição de uma cadeia analítica em três camadas para monitoramento contínuo e online de redes LoRa Mesh; (ii) um modelo de falhas derivado de dados reais de telemetria; (iii) a demonstração empírica da eficácia do LLM como filtro contextual na supressão de alertas não acionáveis; e (iv) o detalhamento de mecanismos de degradação controlada para cenários de indisponibilidade do serviço de validação em nuvem.

2. Trabalhos Relacionados

Os trabalhos correlatos organizam-se em três eixos, sintetizados na Tabela 1: (i) confiabilidade em LoRa e LoRa Mesh; (ii) detecção de anomalias em IoT; e (iii) uso de LLMs no gerenciamento de redes.

2.1. Confiabilidade em LoRa e LoRa Mesh

Cattani et al. (2017), Bor et al. (2016) e Liando et al. (2019) documentaram a severa degradação do LoRa em ambientes e redes densas por multipercurso e colisões, mas limitam-se ao diagnóstico experimental sem propor detecção automática. Solé et al. (2022, 2025) viabilizaram comunicação multi-hop com a LoRaMesher, biblioteca de roteamento adotada neste trabalho, porém sem observabilidade nem tipificação de falhas.

2.2. Detecção de anomalias em IoT e redes sem fio

O Isolation Forest [Liu et al. 2008] consolidou-se como referência não supervisionada sob escassez de dados rotulados. Em LoRa, Senol et al. (2024) aplicam aprendizado federado com foco em privacidade, restrito a ataques específicos e sem filtragem contextual; Aras et al. (2017) catalogam vulnerabilidades de camada física sem arquitetura operacional. Tais abordagens produzem saídas binárias e não distinguem eventos benignos de falhas acionáveis, perpetuando a fadiga de alertas.

2.3. LLMs aplicados ao gerenciamento de redes

Zhou et al. (2025) propuseram o MeshAgent para datacenter, cuja variabilidade de enlace é ordens de magnitude menor que a observada em LoRa. Bonanno et al. (2025) empregam LLMs em mesh para posicionamento indoor, sem tratar detecção de falhas. Nenhum desses trabalhos usa o LLM como filtro semântico sobre detector estatístico em LoRa Mesh.

Tabela 1. Comparação com trabalhos correlatos.

Referência	Tecnologia	Técnica de detecção	LLM	Ambiente	Métricas reportadas
[Liu et al. 2008]	Genérica	Isolation Forest	Não	Benchmarks	AUC

[Cattani et al. 2017]	LoRa	Caracterização empírica	Não	Indoor/Outdoor	PRR, RSSI
[Bor et al. 2016]	LoRaWAN	Modelagem e simulação	Não	Simulação	Taxa de colisão
[Senol et al. 2024]	LoRa	Aprendizado federado	Não	Experimental	Acurácia
[Solé et al. 2022]	LoRa Mesh	Implementação de protocolo	Não	Indoor	EED(End-to-End Delay), PDR (Packet Delivery Ratio)
[Zhou et al. 2025]	Datacenter	Invariantes + LLM	Sim	Simulação	Acurácia
[Bonanno et al. 2025]	Mesh Hermes	BDI + LLM	Sim	Indoor	Precisão espacial

Lacuna identificada. Nenhum trabalho combina detecção estatística e validação semântica por LLM em uma arquitetura unificada para redes LoRa Mesh, orientada à redução de falsos positivos e avaliada em testbed real. É exatamente essa lacuna que o presente artigo busca preencher.

3. Fundamentação Técnica e Científica

Esta seção apresenta os fundamentos que sustentam a cadeia de monitoramento proposta, abrangendo propagação LoRa em ambiente indoor, roteamento mesh, detecção estatística de anomalias e consumo energético.

3.1. Propagação Indoor e Métricas de Enlace LoRa

O transceptor Semtech SX1276[Semtech 2020] opera a 915 MHz com sensibilidade entre −123 dBm (SF7) e −137 dBm (SF12). Em ambientes indoor, à perda de percurso somam-se atenuação de 8 a 15 dB por parede de alvenaria [Rappaport 2002] e desvanecimento multipercurso. Enquanto cenários outdoor apresentam desvio padrão de RSSI tipicamente entre 4 e 13 dB [Goldsmith 2005], Cattani et al. (2017) documentaram alta variabilidade de RSSI em ambientes indoor por efeitos de multipercurso e temperatura. Essa variabilidade é a principal causa de falsos positivos em sistemas baseados em limiares estáticos e justifica o uso de aprendizado de máquina.

3.2. Roteamento em Redes LoRa Mesh

A LoRaMesher[1] [Solé et al. 2022] é uma biblioteca de código aberto para microcontroladores ESP32 acoplados aos transceptores SX1276/SX1278, cujo objetivo é viabilizar redes mesh multi-hop sobre LoRa em dispositivos de baixa potência. A biblioteca abstrai a camada de enlace e oferece roteamento proativo por vetor de distância, descoberta automática de vizinhança, encaminhamento transparente de pacotes e suporte nativo a topologias dinâmicas, características que justificam sua adoção em cenários IoT com restrição de energia e ausência de infraestrutura. Nessa arquitetura, cada nó atua simultaneamente como origem de dados e como relay dos vizinhos, de modo que a indisponibilidade de um nó, seja por falha de hardware ou por esgotamento de bateria, suprime suas rotas de encaminhamento e pode particionar a malha. Os nós emitem periodicamente Hello packets, que atualizam as tabelas de roteamento, funcionam como

[1] LoRaMesher v0.0.11-alpha. Disponível em: https://github.com/LoRaMesher/LoRaMesher. Acesso em: abr. 2026.

heartbeat e servem de fonte passiva para a coleta de métricas de enlace (RSSI, SNR e estabilidade do vizinho). Essa dinâmica introduz o fenômeno de route flapping, em que flutuações de qualidade de sinal provocam alternância repetida entre rotas. Distinguir a autorregeneração legítima da instabilidade topológica (A7) exige a observação contínua da taxa de mudança de rotas em conjunto com o PDR, conforme práticas de histerese consolidadas em protocolos de roteamento para LLN (Low-power and Lossy Networks) [Winter et al. 2012].

3.3 Isolation Forest na Detecção de Outliers

O Isolation Forest (IF) [Liu et al. 2008] fundamenta a Camada 1 da arquitetura. O algoritmo parte do princípio de que anomalias são instâncias minoritárias que tendem a se isolar em menor profundidade média sob partições recursivas aleatórias do espaço de atributos. Cada árvore do ensemble é construída sobre uma subamostra aleatória dos dados; em cada nó, o algoritmo seleciona um atributo e um ponto de corte aleatórios e divide a partição até que cada instância fique isolada em uma folha. A pontuação de anomalia, agregada ao longo da floresta, é dada pela Equação (1):

$$s(x,n) = 2^{-\frac{E[h(x)]}{c(n)}} \quad (1)$$

Na Equação (1), x é a instância avaliada; n é o tamanho da amostra de cada árvore; E[h(x)] é a profundidade média de isolamento de x ao longo da floresta; e c(n) é um fator de normalização baseado no comprimento esperado de caminho em uma árvore binária de busca não balanceada. Valores de s(x, n) próximos de 1 indicam forte evidência de anomalia; valores próximos de 0 caracterizam operação típica.

Na implementação com scikit-learn[2], o método score_samples() retorna o simétrico de s(x, n), de modo que valores mais negativos correspondem a maior grau de anomalia. Dois hiperparâmetros governam o comportamento do modelo: o número de árvores, que regula a estabilidade estatística do ensemble, e o parâmetro contamination, que representa a fração esperada de outliers no conjunto de treino e calibra o limiar interno entre operação típica e anomalia. Os valores adotados neste trabalho são especificados na Seção 4.2.1.

O IF foi adotado pela compatibilidade com o regime de inferência contínua em infraestrutura de recursos limitados. O algoritmo é indutivo e opera com subamostragem: o treinamento processa subconjuntos fixos de amostras por árvore, tornando o custo de treino e de inferência independente do tamanho total do histórico, o que mantém o ciclo periódico de detecção viável em máquinas de recursos modestos. Abordagens baseadas em densidade local, como o Local Outlier Factor (LOF), são transdutivas: para classificar cada nova janela de 15 minutos, seria necessário refitar o modelo sobre toda a base histórica. Esse comportamento é incompatível com o ciclo de detecção de 60 s adotado na arquitetura proposta, pois o custo de reprocessamento cresceria continuamente à medida que novos dados chegam. Acrescenta-se ainda a instabilidade do LOF diante de janelas com poucos pontos e vetores multivariados de alta dimensão. A limitação do

[2] scikit-learn 1.6.1. Disponível em: https://scikit-learn.org. Acesso em: abr. 2026.

Isolation Forest de operar sobre o desvio estatístico sem indicar causalidade é compensada pelas camadas subsequentes da arquitetura.

4. Arquitetura Proposta

A arquitetura proposta para detecção de anomalias e falhas topológicas é modular, não intrusiva e opera de forma contínua e online, com latência da ordem de dezenas de segundos, integrando a coleta de métricas na borda com o processamento analítico em nuvem. A Figura 1 ilustra a topologia física e a comunicação entre os dispositivos finais, o gateway e os serviços de backend.

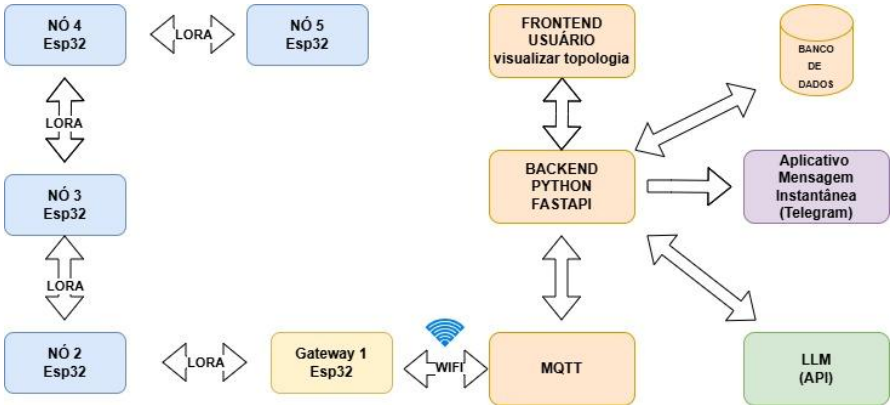

Figura 1. Arquitetura do sistema para detecção de anomalias em rede mesh LoRa

4.1. Infraestrutura de Rede e Telemetria

Conforme a Figura 1, a cada ciclo de roteamento a infraestrutura intercepta os pacotes de controle trocados entre os nós ESP32 e extrai métricas de saúde: RSSI, SNR, PDR, variação de tensão da bateria e estabilidade do nó pai. Esses dados trafegam via LoRa em múltiplos saltos até o Gateway 1, que encaminha a telemetria por WiFi ao broker MQTT[3] hospedado na nuvem. O backend, desenvolvido em Python[4] com FastAPI[5] e PostgreSQL[6], centraliza a recepção contínua e orquestra a cadeia analítica da Seção 4.2. Três integrações complementam a arquitetura: o frontend web para visualização da topologia, a API do LLM para validação semântica e o Telegram[7] para entrega de alertas ao operador.

4.2. Pipeline Analítico de Detecção de Anomalias

Processar dados brutos de IoT diretamente com LLMs é inviável pelo alto custo, latência e limites de tokens. Para contornar essa restrição, a arquitetura estrutura o fluxo em três camadas analíticas sucessivas, ilustradas na Figura 2.

O ciclo inicia com a leitura da telemetria do banco de dados PostgreSQL, agregada em janelas de 15 minutos, da qual se extraem 11 features multivariadas. Esse vetor

[3] MQTT versão 3.1.1. OASIS Standard, 2014. Disponível em: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html
[4] Python versão 3.13.1. Python Software Foundation, 2024. Disponível em: https://www.python.org
[5] FastAPI versão 0.115.6. S. Ramírez, 2024. Disponível em: https://fastapi.tiangolo.com
[6] PostgreSQL 16. Disponível em: https://www.postgresql.org. Acesso em: abr. 2026.
[7] Telegram Bot API. Disponível em: https://core.telegram.org/bots/api. Acesso em: abr. 2026.

alimenta a Camada 1 (Isolation Forest): amostras com escore negativo avançam como candidatas a anomalia; as demais são descartadas. As retidas seguem para a Camada 2, onde regras determinísticas realizam a tipificação causal em uma das sete classes (A1 a A7). Por fim, a Camada 3 executa a validação semântica pelo LLM, combinando o vetor tipificado com o histórico dos últimos 10 minutos, o estado atual da topologia e a base de conhecimento da aplicação, conforme indicado na Figura 2. O veredito determina o desfecho: ocorrências REAL e WARNING disparam alerta via Telegram; falso positivo e inconclusivo são registrados no log de auditoria.

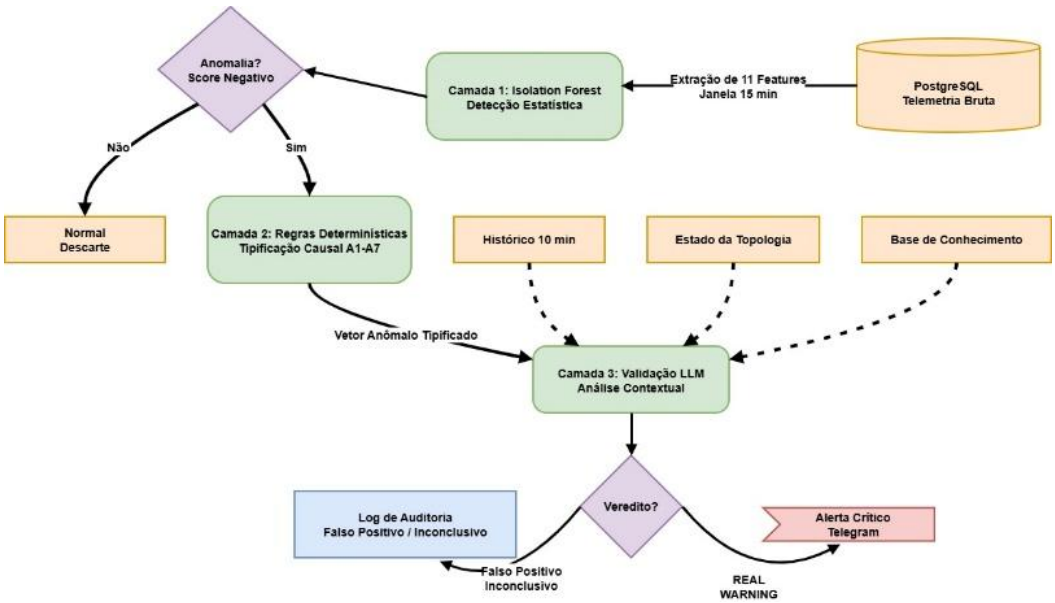

Figura 2. Fluxo de processamento hierárquico em três camadas

4.2.1. Camada 1: Detecção Estatística de Outliers (Isolation Forest)

A Camada 1 aplica o Isolation Forest (detalhado na Seção 3.3) a janelas de 15 minutos, extraindo o vetor de 11 features da Tabela 2. O objetivo é exclusivamente sinalizar outliers estatísticos, sem explicar ou classificar a causa da falha.

Tabela 2. Features extraídas para detecção (Camada 1)

#	Feature	Unid.	Descrição
1	rssi_mean	dBm	Média da força do sinal (RSSI)
2	rssi_std	dB	Desvio padrão do RSSI (indica instabilidade)
3	snr_mean	dB	Média da relação sinal-ruído
4	snr_std	dB	Desvio padrão do SNR
5	pdr_mean	0–10	Índice determinístico de entrega de pacotes
6	battery_delta	%	Variação (descarga) de tensão da bateria
7	packet_loss_rate	0–1	Taxa de perda calculada por gaps de sequência
8	rssi_snr_divergence	dB	Divergência absoluta entre variâncias de RSSI e SNR
9	beacon_interval_std	s	Irregularidade temporal na emissão de beacons
10	route_change_rate	0–1	Fração de mudanças de nó pai (parent) observadas
11	link_asymmetry	dBm	Assimetria bidirecional média do enlace de rádio

O parâmetro de contaminação foi fixado em 0,08, estimativa conservadora da fração de outliers esperada no tráfego normal da malha. A floresta foi configurada com 100 árvores de isolamento, valor que oferece boa convergência estatística sem custo computacional relevante para o perfil de dados do testbed.

4.2.2. Camada 2: Tipificação Causal Determinística

Os dados sinalizados pela Camada 1 avançam para regras determinísticas estritas que mapeiam a anomalia em uma das sete classes causais (A1–A7, detalhadas na Seção 5). Enquanto a Camada 1 detecta o desvio, a Camada 2 rotula a sua natureza.

4.2.3. Camada 3: Validação Contextual Avançada por LLM

Em ambiente indoor, a simples movimentação de pessoas pode gerar flutuações de sinal que a Camada 1 sinaliza como anomalias. A Camada 3 atua como validador semântico e filtro final do fluxo, por meio de um LLM acessado via API. Frente a regras determinísticas adicionais, modelos supervisionados clássicos e TinyML embarcado, o LLM reúne três atributos exigidos pelo problema: raciocínio causal sobre dados heterogêneos, operação few-shot ou zero-shot sem fine-tuning e explicação legível ao operador. Regras estendidas exigiriam codificar previamente todos os cenários contextuais, inviável dada a variabilidade indoor.

A validação semântica foi realizada com o modelo Claude Haiku (claude-haiku-4-5-20251001, Anthropic[8]), acessado via API com temperatura padrão. A escolha pelo modelo Haiku justifica-se pelo custo por chamada, latência compatível com o ciclo de 60 s e capacidade de raciocínio contextual adequada à classificação das anomalias definidas.

Em operação, o modelo recebe um prompt estruturado com: (i) métricas e tipo da anomalia; (ii) histórico de 10 minutos do nó; (iii) estado global da malha e nós offline; e (iv) histórico de decisões anteriores. O LLM retorna um JSON com o veredito (REAL, WARNING, FALSO_POSITIVO ou INCONCLUSIVO) e a explicação causal. Apenas falhas confirmadas acionam a notificação ao operador.

5. Modelo de Falhas e Catálogo de Anomalias

Para que um sistema de tolerância a falhas seja eficaz, os desvios estatísticos detectados na subcamada de rede precisam ser mapeados para causas físicas ou lógicas compreensíveis. Essa responsabilidade é delegada à Camada 2 (Tipificação Determinística), que classifica as anomalias candidatas em sete classes operacionais (A1–A7), com seus critérios de ativação, fenômenos associados e justificativas científicas consolidados na Tabela 3.

Tabela 3. Catálogo do Modelo de Falhas e Anomalias (A1–A7)

Classe	Fenômeno	Categoria	Limiar de Ativação	Features	Descrição Físico-Lógica e Justificativa Científica
A1	Degradação de Sinal	Rádio	RSSI < −100 dBm ∨ (rssi_std > 10 ∧ SNR < 0 dB)	rssi_mean, rssi_std, snr_mean, snr_std	Limiar preditivo com ~23 dB de margem sobre a sensibilidade do SX1276 em SF7 (−123 dBm) [Semtech 2020]; PDR degrada ao aproximar-se da

[8]Anthropic (2025). Claude Haiku. Disponível em: https://www.anthropic.com/claude. Acesso em: abr. 2026.

					sensibilidade nominal [Augustin et al. 2016].
A2	Perda de Pacotes	Rádio	packet_loss > 0,30 ∨ rx_count < 2	packet_loss_rate, pdr_mean	Quedas abruptas e colapso de entrega indicam falha crítica; limiar de 30% alinhado a estudos de contenção LoRa [Bor et al. 2016] e degradação indoor sem LOS [Liando et al. 2019].
A3	Anomalia Energética	Hardware	battery_delta < −20 %/janela	battery_delta	Queda de ≥ 20 pontos percentuais em uma janela de 15 min representa ≈ 8× a descarga ociosa típica [Casals et al. 2017].
A4	Link Instável	Topologia	rssi_std > 8 ∧ snr_std > 5 ∧ rx_count < 4	rssi_std, snr_std, pdr_mean	Caracteriza fast fading severo; conjunção de alta variância com degradação de SNR/PDR distingue instabilidade real de multipercurso benigno [Cattani et al. 2017; Goldsmith 2005].
A5	Interferência (Jamming)	Segurança	(div > 5 ∧ SNR < 3 ∧ rx_count < 3) ∨ (RSSI > −100 dBm ∧ snr_std > 5 ∧ rx_count < 2)	rssi_snr_divergence, snr_std, pdr_mean, beacon_interval_std	Divergência entre SNR decrescente e RSSI estável é assinatura de jamming reativo [Xu et al. 2005; Xu et al. 2006]; viabilidade em LoRa com hardware COTS demonstrada por [Aras et al. 2017].
A6	Degradação de Hardware	Hardware	link_asymmetry > 15 dB	link_asymmetry, rssi_std	Canal RF é simétrico [Balanis 2016]; assimetria >15 dB indica falha física unilateral (ex.: antena TX).
A7	Instabilidade Topológica	Topologia	route_change_rate > 0,30	route_change_rate, rssi_std, pdr_mean	Route flapping agrava a LLN [Winter et al. 2012] com overhead de controle, aumento de latência e reordenação de pacotes.

6. Confiabilidade e Tolerância a Falhas

Esta seção descreve os mecanismos que asseguram a operação contínua da arquitetura proposta, abrangendo a execução periódica, as propriedades de confiabilidade decorrentes da separação em camadas e os procedimentos de degradação controlada diante da indisponibilidade do LLM.

6.1. Operação Periódica

O fluxo de detecção executa em ciclos de 60 segundos em uma thread dedicada, independente da API principal. A cada ciclo, o sistema percorre todas as redes ativas e, para cada nó com dados suficientes, executa em sequência: extração de features, inferência pelo Isolation Forest, tipificação causal (A1 a A7), deduplicação de alertas, persistência no banco de dados, difusão via WebSocket e submissão à validação por LLM.

6.2. Confiabilidade da Arquitetura

A arquitetura mantém operação confiável por quatro mecanismos: (i) separação em camadas, de modo que a indisponibilidade da Camada 3 não interrompe as Camadas 1 e 2; (ii) retreinamento periódico com janelas deslizantes e histórico-alvo de 24 horas, acompanhando mudanças graduais do ambiente; (iii) treinamento progressivo, com operação preliminar a partir de 10 amostras; e (iv) trilha de auditoria com features, score, classificação e, quando disponível, parecer do LLM com justificativa, viabilizando rastreabilidade pós-incidente.

6.3. Degradação Controlada e Fail-Safe

Conforme descrito na Seção 4.2.3, a Camada 3 introduz uma dependência externa que não pode se tornar ponto único de falha. Toda comunicação com o LLM é encapsulada em chamadas não bloqueantes com timeouts estritos. Se a API exceder o tempo limite ou retornar erro, o sistema adota postura de falha aberta: em vez de silenciar os alertas, contorna a Camada 3 e despacha diretamente as anomalias de severidade alta com base na tipificação da Camada 2.

7. Testbed e Resultados Experimentais

Esta seção apresenta a avaliação experimental da arquitetura em um testbed real com cinco nós ESP32. São descritos a configuração do ambiente, o perfil de falsos positivos gerados pela detecção base, a eficácia da validação contextual pelo LLM e as métricas de resiliência e latência do sistema.

7.1 Configuração do Testbed em Ambiente Real

Cinco nós Heltec[9] V2 LoRa (ESP32 + transceptor SX1276) a 915 MHz foram posicionados com obstrução por paredes em condição de visada não direta, conforme a Figura 3. Os nós operam com Spreading Factor fixo em SF7 e largura de banda de 125 kHz; o controle de topologia segue o roteamento proativo da LoRaMesher. A rede formou rotas multi-hop em que nós periféricos utilizam intermediários como relays para alcançar

[9] Heltec V2. WiFi LoRa 32 (V2) — Schematic Diagram. Disponível em: https://resource.heltec.cn/download/WiFi_LoRa_32/WiFi%20Lora32.pdf. Acesso em: abr. 2026.

o gateway central, ilustrado na Figura 4. Durante a avaliação, o backend processou assincronamente 38.308 pacotes de telemetria.

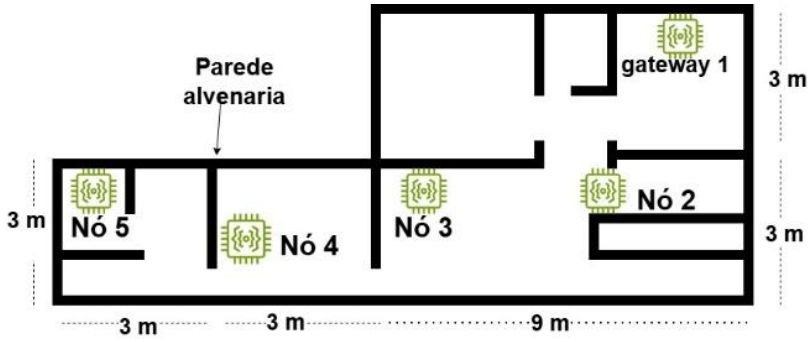

Figura 3. Layout da disposição dos nós no ambiente indoor.

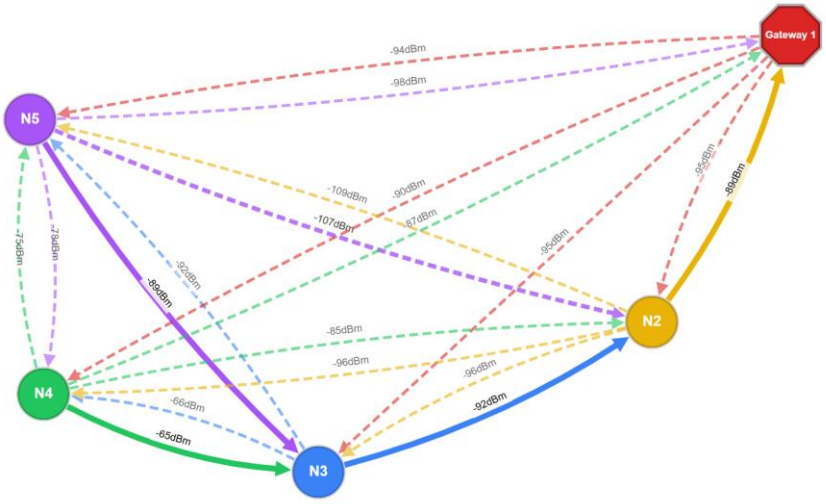

Figura 4. Grafo em tempo real da topologia multi-hop (rotas ativas em linhas contínuas e vizinhos em tracejadas).

7.2 Perfil do Canal e a Geração de Falsos Positivos

A Camada 1 sinalizou 156 anomalias estatísticas ao longo do período de avaliação. Após a tipificação pela Camada 2, a distribuição causal revelou: 148 ocorrências de Degradação de Sinal (A1), 6 de Perda de Pacotes (A2) e 2 de Degradação de Hardware TX (A6). O volume de alertas A1 confirma empiricamente o problema da fadiga de alertas: em um sistema tradicional, o operador seria notificado cerca de 50 vezes por dia apenas por reflexões de sinal ou abertura de portas, tornando o monitoramento operacionalmente inviável.

7.3 Eficácia da Validação Contextual (Camada 3)

As 156 anomalias foram submetidas ao LLM para avaliação do contexto holístico da rede. A Tabela 4 apresenta a distribuição dos vereditos.

Tabela 4. Distribuição de vereditos da Camada 3 (LLM) para 156 anomalias.

Veredito	Quantidade	% do Total	Ação do Sistema
FALSO_POSITIVO	95	60,9%	Alerta suprimido
INCONCLUSIVO	44	28,2%	Alerta suprimido
REAL	12	7,7%	Notificação Crítica
WARNING	5	3,2%	Notificação de Atenção

A Camada 3 suprimiu autonomamente 89,1% dos falsos alertas, encaminhando ao operador apenas 17 eventos acionáveis. A correção dos 95 vereditos FALSO_POSITIVO foi confirmada por auditoria dos registros: todos apresentavam RSSI médio de −84,6 dBm e PDR de 92,3%, acima dos limiares críticos do A1, evidenciando que o Isolation Forest sinalizou variações benignas do canal sem degradação operacional real. Nos casos REAL, o RSSI estava persistentemente abaixo de −103 dBm com SNR degradado e padrão temporal sustentado. Os 44 casos INCONCLUSIVO decorreram de falha técnica de integração: o LLM produziu vereditos válidos em todos, porém em formato que o sistema não conseguiu interpretar. O mecanismo de degradação controlada (Seção 6.3) suprime esses alertas por padrão, sem impacto ao operador.

Em um caso de falso positivo por multipercurso, o LLM suprimiu o alerta ao constatar RSSI de −56 dBm com SNR e PDR excelentes, caracterizando apenas variância estatística. Em outro evento, o modelo correlacionou alta variância de RSSI com SNR estável e com o histórico de vereditos anteriores do nó, reclassificando definitivamente como falha real. Essa capacidade de rastrear o histórico supera abordagens baseadas em limiares estáticos.

7.4 Métricas do Isolation Forest com Ground Truth Sintético

Para avaliar a Camada 1 em isolamento, adotou-se injeção controlada de falhas sobre a telemetria persistida, isolando o classificador do pipeline físico. Quatro classes foram induzidas em nós distintos (A3 em No4, A4 em No2, A5 em No3, A7 em No5), com registros sintéticos calibrados para satisfazer os critérios da Camada 2 (Tabela 3). A classe A6 foi excluída pela exigência de registros bidirecionais da feature link_asymmetry. Por contar com uma única indução por classe, os resultados são indicadores sob condição controlada, não estimativas estatisticamente generalizáveis. A Tabela 5 apresenta os resultados. O Isolation Forest atingiu Recall de 1,00 e Taxa de Falsos Negativos de 0,00 em todas as classes; a Precision variou de 0,25 (A4) a 0,50 (A5).

Tabela 5. Métricas de detecção do Isolation Forest por classe de anomalia induzida.

Anomalia	Precision	Recall	F1-Score	Taxa FN
A3	0,33	1,00	0,50	0,00
A4	0,25	1,00	0,40	0,00
A5	0,50	1,00	0,67	0,00
A7	0,33	1,00	0,50	0,00
Global	0,33	1,00	0,50	0,00

A Precision reduzida tem duas origens: co-detecções de A2 dentro das janelas de injeção, artefato esperado do modelo e não característica física, e seis eventos reais não rotulados fora das janelas (cinco A3 e um A7) que entram como FP no cálculo, configurando cota inferior pessimista do desempenho real. Os resultados confirmam o papel projetado para a Camada 1: triagem de alta sensibilidade, sem falsos negativos, com Precision corrigida pelas camadas subsequentes.

7.5 Avaliação de Resiliência e Latência

A tolerância a falhas da arquitetura foi validada pela estabilidade da integração entre nuvem e borda. Todas as 156 anomalias foram processadas pela API da Camada 3 sem registro de erros de indisponibilidade. A latência introduzida pela validação contextual

mostrou-se compatível com o ciclo analítico assíncrono de 60 segundos, não impondo restrições ao regime de operação adotado. O processo do backend consumiu em média 10% de CPU e 135 MB de RAM no VPS de 2 vCPUs e 1 GB de memória, valores obtidos por amostragem periódica do processo Docker durante o ciclo de detecção ativo, sem carga de injeção sintética simultânea, sem uso de GPU ou acelerador de hardware, confirmando a viabilidade de operação contínua em infraestrutura de baixo custo.

8. Discussão e Trabalhos Futuros

Esta seção interpreta os resultados com base nas hipóteses que motivaram a arquitetura em três camadas, discute como cada etapa contribuiu para a redução de falsos positivos e examina as limitações do estudo e as direções de trabalho futuro.

8.1. A Taxa de Falsos Positivos e o Papel do Ambiente Indoor

Os resultados da Seção 7.2 confirmam o que a literatura de propagação indoor antecipa (Seção 3.1): das 156 anomalias detectadas, 148 corresponderam a flutuações de sinal (A1), equivalentes a cerca de 50 alertas diários atribuíveis a eventos benignos como trânsito de pessoas ou fechamento de portas. Cabe ressalva metodológica: o testbed operou inteiramente em ambiente com bloqueio por paredes de alvenaria, sem cenário pareado em linha de visada (LOS). Essa ausência impede isolar quantitativamente o peso da atenuação das paredes frente a outras fontes de variabilidade (interferência no espectro, variações térmicas e oscilações de alimentação). Ainda assim, o volume observado evidencia que um modelo estatístico puro não distingue obstáculo físico permanente de transitório, limitação que conduz equipes de operação a ignorarem sistemas de monitoramento. A filtragem contextual da Camada 3 foi, portanto, uma exigência arquitetural, não um refinamento opcional.

8.2. Eficácia e Limitações da Validação por LLM

A escolha do LLM na Camada 3 apoia-se em quatro propriedades do problema: (i) ausência de conjunto rotulado em volume suficiente para treino supervisionado; (ii) necessidade de raciocínio multivariado sobre sinais heterogêneos, em que modelos pré-treinados operam de forma competitiva em regime few-shot ou zero-shot, sem necessidade de fine-tuning [Brown et al. 2020]; (iii) necessidade de explicabilidade para o operador, já que o LLM acompanha cada veredito com uma justificativa em linguagem natural; e (iv) restrições de hardware do backend, que opera em VPS com 1 GB de RAM, inviabilizando hospedagem local de LLMs. O catálogo de anomalias pode ser expandido por simples ajustes no prompt, sem ciclos de rotulagem.

8.3. Limitações e Trabalhos Futuros

Cinco limitações delineiam a agenda futura. Primeiro, o testbed foi conduzido em um único ambiente indoor; replicações em ambientes outdoor e em cenários fisicamente distintos, como ambientes urbanos e rurais, estão previstas. Segundo, a ausência de cenário pareado em linha de visada direta impede a atribuição estrita da variabilidade de canal à atenuação por paredes, motivando um estudo futuro com nós posicionados simultaneamente em condições de visada direta e obstruída. Terceiro, a dependência de LLM em nuvem introduz latência, custo por chamada e requisito de conectividade; pretende-se avaliar modelos locais de menor porte, comparando supressão de falsos positivos e custo operacional. Quarto, o custo computacional da arquitetura não foi

mensurado de forma sistemática; sua caracterização está prevista, inclusive em contraste com soluções puramente determinísticas. Quinto, a Camada 1 utiliza exclusivamente o Isolation Forest; avaliar a combinação deste com outros algoritmos de detecção de anomalias ou com redes neurais pode ampliar a precisão da triagem sem comprometer o recall obtido.

9. Conclusão

O gerenciamento de falhas em redes IoT de borda sofre cronicamente com o excesso de alarmes falsos gerados pela instabilidade do meio físico. Este trabalho propôs e validou uma arquitetura de monitoramento em três camadas voltada a prover alta observabilidade, mitigação da fadiga de alertas e degradação controlada em redes LoRa Mesh.

A combinação de detecção estatística por Isolation Forest com tipificação determinística permitiu identificar de forma contínua degradações de rede sem necessidade de dados rotulados. A introdução do LLM como supervisor semântico na Camada 3 demonstrou eficácia substancial em campo, suprimindo 89,1% dos alertas não acionáveis. A escolha apoia-se na ausência de conjuntos rotulados em escala suficiente, no raciocínio multivariado sobre sinais heterogêneos e na viabilidade de consumo via API, que dispensa infraestrutura de treinamento local.

Os resultados confirmam que a inteligência artificial generativa, devidamente encapsulada por mecanismos de degradação controlada e limites de tempo estritos, pode integrar a cadeia de tolerância a falhas de infraestruturas IoT críticas sem sobrecarregar o operador. A avaliação por indução controlada de falhas (Seção 7.4) demonstrou que o Isolation Forest atinge Recall de 1,00 para as classes A3, A4, A5 e A7, confirmando sua adequação como triagem de alta sensibilidade na Camada 1.

Agradecimentos

Os autores agradecem ao Centro de Excelência em Redes Inteligentes Sem Fio e Serviços Avançados (CERISE), CAPES e FAPEG pelo apoio e financiamento à pesquisa.

Referências

Aras, E., Ramachandran, G. S., Lawrence, P. and Hughes, D. (2017) "Exploring the Security Vulnerabilities of LoRa", in Proc. 3rd IEEE Int. Conf. Cybernetics (CYBCONF), Exeter, UK, pp. 1–6.

Augustin, A., Yi, J., Clausen, T. and Townsley, W. M. (2016) "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things", Sensors, vol. 16, no. 9, p. 1466.

Balanis, C. A. (2016) Antenna Theory: Analysis and Design, 4th ed., Wiley, Hoboken, NJ, USA.

Bonanno, M., Russo, M., Santoro, C., Santoro, F. F. and Tudisco, A. (2025) "BDI-Driven Indoor Positioning and Assistance via Hermes Mesh Networks and LLM Interfaces", in Proc. 23rd IEEE Int. Conf. Pervasive Intelligence and Computing (PICom).

Bor, M., Roedig, U., Voigt, T. and Alonso, J. M. (2016) "Do LoRa Low-Power Wide-Area Networks Scale?", in Proc. 19th ACM Int. Conf. Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), Malta, pp. 59–67.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P. et al. (2020) "Language Models are Few-Shot Learners", in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877–1901.

Casals, L., Mir, B., Vidal, R. and Gomez, C. (2017) "Modeling the Energy Performance of LoRaWAN", Sensors, vol. 17, no. 10, p. 2364.

Cattani, M., Boano, C. A. and Römer, K. (2017) "An Experimental Evaluation of the Reliability of LoRa Long-Range Low-Power Wireless Communication", J. Sensor Actuator Networks, vol. 6, no. 2, p. 7.

Goldsmith, A. (2005) Wireless Communications, Cambridge University Press, Cambridge, UK.

Liando, J. C., Gamage, A., Tengourtius, A. W. and Li, M. (2019) "Known and Unknown Facts of LoRa: Experiences from a Large-Scale Measurement Study", ACM Trans. Sensor Networks, vol. 15, no. 2, pp. 1–35.

Liu, F. T., Ting, K. M. and Zhou, Z.-H. (2008) "Isolation Forest", in Proc. 8th IEEE Int. Conf. Data Mining (ICDM), Pisa, Italy, pp. 413–422.

Rappaport, T. S. (2002) Wireless Communications: Principles and Practice, 2nd ed., Prentice Hall, Upper Saddle River, NJ, USA.

Semtech Corporation (2020) "SX1276/77/78/79 Datasheet", Rev. 7. Available: https://www.semtech.com/products/wireless-rf/lora-connect/sx1276.

Senol, N. S., Baza, M., Rasheed, A. and Alsabaan, M. (2024) "Privacy-Preserving Detection of Tampered Radio-Frequency Transmissions Utilizing Federated Learning in LoRa Networks", Sensors, vol. 24, no. 22, p. 7336.

Solé, J. M., Centelles, R. P., Freitag, F. and Meseguer, R. (2022) "Implementation of a LoRa Mesh Library", IEEE Access, vol. 10, pp. 113158–113171.

Solé, J. M., Freitag, F., Navarro, L. and Selimi, M. (2025) "Demonstration of Concurrent Application Provision with LoRaMesher", in Proc. 7th Experiment@ International Conference (exp.at'25), Faial, Azores, Portugal, pp. 231.

Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J. P. and Alexander, R. (2012) "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, IETF.

Xu, W., Ma, K., Trappe, W. and Zhang, Y. (2006) "Jamming Sensor Networks: Attack and Defense Strategies", IEEE Network, vol. 20, no. 3, pp. 41–47.

Xu, W., Trappe, W., Zhang, Y. and Wood, T. (2005) "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks", in Proc. 6th ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc), Urbana-Champaign, IL, USA, pp. 46–57.

Zhou, Y., Hsieh, K., Mani, S. K., Kandula, S. and Liu, Z. (2025) "MeshAgent: Enabling Reliable Network Management with Large Language Models", Proc. ACM Meas. Anal. Comput. Syst., vol. 9, no. 3, Art. 52.