LoRaDoctor: LLM-Driven Diagnosis and Adaptive Policy Optimization for Reducing Packet Error Rate in LoRaWAN Networks

Dongyi Ma[†*], Aitian Ma[‡*], Sirui Luo[†], Martin de Jode[†], Andrew Hudson-Smith[†], Mo Sha[‡]
[†]Bartlett Centre for Advanced Spatial Analysis, University College London
[‡]Knight Foundation School of Computing and Information Sciences, Florida International University
[†]{dongyi.ma.21, sirui.luo.24, m.dejode, a.hudson-smith}@ucl.ac.uk, [‡]{aima, msha}@fiu.edu

***Abstract*—LoRaWAN is widely used for large-scale Internet of Things (IoT) deployments, but real-world reliability is often affected by high packet error rate. Existing optimization methods, such as heuristics or supervised learning, cannot fully capture the effects of environment, spatial layout, and network dynamics, which limits their adaptability. In this paper, we present LORADOCTOR, the first framework that leverages Large Language Models (LLMs) to optimize LoRaWAN. LORADOCTOR performs causal analysis, generates adaptive transmission policies, and predicts network performance in an interpretable way. We perform simulation-based evaluations using a year-long dataset collected from eight sensors deployed in East London. The results show that LORADOCTOR can significantly reduce packet error rates compared to both the default LoRaWAN settings and standard machine learning methods. Our evaluation also identifies distance-based path loss, temperature effects, and human activity as the main causes of packet error rate, showing its potential to support more reliable and adaptive LoRaWAN deployments in future urban environments.**

***Index Terms*—LoRaWAN, Low-Power Wide Area Networks, Packet Error Rate Reduction, Causal Analysis, Adaptive Policy Optimization, Large Language Models, Multi-Modal Reasoning**

I. Introduction

Low-Power Wide Area Networks (LPWANs) serve as the communication backbone for the rapidly expanding Internet of Things ecosystem, with LoRaWAN deployments alone expected to connect over 200 million devices by 2025 [1]. These networks enable critical applications ranging from smart city infrastructure and environmental monitoring to industrial automation, where reliable data transmission is essential for operational safety and efficiency [2], [3]. According to the study conducted by Ma et al. [4], bespoke LoRaWAN heat sensors can be used to explore microclimate effects within London's urban heat islands. However, real-world LoRaWAN deployments face a fundamental challenge: substantial packet error rate in dense urban environments, severely compromising network reliability and rendering many Internet of Things (IoT) applications impractical [5], [6], [7], [8], [9].

Existing optimization methods face three fundamental limitations that prevent effective adaptation to real-world conditions. First, heuristic-based approaches like Adaptive Data Rate (ADR) algorithms rely on simplified models that assume network conditions and fail to capture the complex interdependencies between multiple performance factors [10]. Studies show that default ADR achieves packet delivery ratios below 60% in dense deployments, highlighting the inadequacy of rule-based optimization [11]. Second, supervised machine learning approaches require extensive labeled training data for each deployment scenario and cannot generalize across different environmental conditions or urban topologies [12], [13]. Third, existing approaches lack interpretability, making it impossible for network operators to understand optimization decisions or adapt to changing deployment requirements [14].

We observe that Large Language Models (LLM) offer unique capabilities for multi-factor reasoning and causal inference that directly address these limitations. Unlike traditional optimization methods, LLMs can analyze heterogeneous data sources. These include network telemetry, environmental measurements, spatial characteristics, and temporal patterns. They can identify causal relationships across these factors. Based on this analysis, LLMs generate targeted and context-aware optimization policies [15], [16]. Moreover, LLMs provide human-interpretable explanations for optimization decisions, enabling network operators to understand system behavior and validate policy recommendations.

We present LORADOCTOR, the first LLM-driven framework for LoRaWAN network optimization that combines causal analysis, adaptive policy generation, and performance prediction. The framework addresses three core technical challenges: (1) multi-factor data integration and causal analysis across heterogeneous data sources, (2) constraint-aware policy generation that complies with LoRaWAN regulatory requirements and hardware limitations, and (3) performance prediction and validation through LLM-driven simulation. Our main contributions are as follows.

- We develop the first LLM-based method for LoRaWAN, which identifies causal links between environmental, spatial, and network factors. The approach combines statistical checks with physics-based reasoning.

*The first two authors contributed equally to this work.

- We build a constraint-aware optimization engine, which creates clear policies that adjust transmission settings, duty cycles, and retransmissions. The system reacts to real-time conditions and stays within regulations.
- We propose a new simulation method powered by LLM reasoning, which predicts how policies will perform in various scenarios. This avoids the need for detailed mathematical radio models.

Our paper is organized as follows. Section II reviews the related work. Section III presents our design of LORADOCTOR. Section IV describes our collected data and analysis. Section V evaluates LORADOCTOR. Section VI concludes this paper.

II. Related Work

LoRaWAN optimization research spans three main areas: heuristic-based approaches, machine learning (ML)-driven methods, and emerging AI-based network management. While each has advanced the state of the art, existing solutions struggle with the complex, multi-factor nature of packet error rate in real deployments.

A. Heuristic-Based Optimization

Heuristic parameter tuning based on distance and coverage is common, but fails in dynamic environments. Adelantado et al. [10] show default ADR can drop delivery ratios below 60% in dense networks. Cuomo et al. [11] improve capacity by 40% with EXPLoRa, and Zorbas et al. [17] reduce collisions by 25% via interference-aware channel selection, but both rely on fixed assumptions and manual tuning.

B. ML-Based Optimization

ML methods adapt parameters dynamically with better performance. Sandoval et al. [12] use Q-learning to cut energy use by 30% while maintaining high delivery rates, but require long training. Xu et al. [13] predict optimal settings from environmental features, improving success rates by 45%, yet lack causal reasoning. Chen et al. [14] achieve 60% packet error rate reduction in simulation using deep networks, but their method is non-interpretable.

C. AI-Driven Network Management

LLMs have been applied to enterprise network configuration [18] and diagnosis [19], while Zhao et al. [20] optimize cellular handovers with transformers. However, these works target high-power networks, overlooking LPWAN constraints such as ultra-low energy use and duty-cycle limits.

D. Gap and Contribution

Current approaches lack (1) multi-factor reasoning across environmental, spatial, and temporal factors, (2) causal inference to identify root loss causes, and (3) interpretable decision-making for operator trust. We present the first LLM-driven LoRaWAN optimizer integrating multi-factor analysis, causal reasoning, and adaptive policy generation, validated with extensive real-world data.

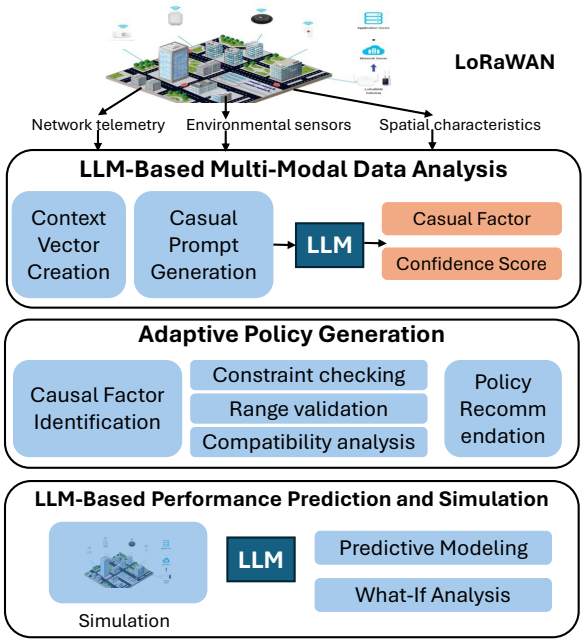

Fig. 1: System Architecture.

III. Methodology

Our LLM-driven LoRaWAN optimization framework consists of four integrated components that work together to analyze network performance, identify root causes of packet error rate, generate adaptive policies, and predict optimization effectiveness. Figure 1 presents the overall system architecture, illustrating the data flow from real-time network monitoring through LLM-based analysis[1] to policy implementation and performance feedback.

A. Problem Formulation

We formulate LoRaWAN network optimization as an optimization problem aimed at minimizing Packet Error Rate (PER). Let $S = \{s_1, s_2, \ldots, s_n\}$ represent the set of n sensors in the network, where each sensor s_i is characterized by its spatial location, environmental conditions, and operational parameters.

For each sensor s_i, the packet error rate $PER_i(t)$ at time t is influenced by a comprehensive set of factors:

$$F_i(t) = \{d_i, T_i(t), H_i(t), A_i(t), P_i(t)\} \tag{1}$$

where d_i represents the distance from sensor s_i to the gateway, $T_i(t)$ denotes the ambient temperature at time t, $H_i(t)$ represents the humidity level at time t, $A_i(t)$ captures the human activity level affecting the sensor environment, and $P_i(t)$ encompasses the transmission parameters, including spreading factor $SF_i(t)$, transmission power $TX_{power,i}(t)$, and duty cycle $DC_i(t)$.

The optimization objective is to determine the optimal parameter configuration $P_i^*(t)$ that minimizes PER:

$$\min_{P_i(t)} PER_i(t) \tag{2}$$

[1]We use ChatGPT-4o as the LLM backbone in this work.

subject to LoRaWAN regulatory constraints, transmission power limitations, and duty cycle restrictions.

The primary challenge lies in modeling the complex, nonlinear relationships between environmental factors and packet error rate. These dependencies exhibit temporal variations and cross-correlations that traditional optimization approaches struggle to capture effectively, necessitating advanced machine learning or adaptive optimization techniques to achieve optimal network performance.

B. System Overview

As Figure 1 shows, the architecture integrates heterogeneous data sources from LoRaWAN deployments, including network telemetry, environmental sensors, and spatial information. An LLM-based multi-modal analysis engine extracts causal factors and confidence scores by combining context vectors and prompt generation. These insights drive an adaptive policy generation module, which performs constraint checking, range validation, and compatibility analysis to recommend optimized transmission policies. Finally, an LLM-powered simulation and performance prediction layer enables predictive modeling and what-if analysis, ensuring robust policy evaluation before real-world deployment.

C. LLM-Based Multi-factor Data Analysis

The foundation of our approach is a multi-factor data analysis component that leverages LLMs to understand complex relationships between diverse data sources. Unlike traditional machine learning approaches that require extensive feature engineering and labeled training data, our LLM-based analyzer can reason about causal relationships and identify patterns across heterogeneous data types.

1) Data Integration and Preprocessing: Our system continuously collects data from multiple sources: network telemetry (Received Signal Strength Indicator (RSSI) measurements, Signal-to-Noise Ratio (SNR), and frame counters), environmental sensors (temperature and humidity readings), and spatial characteristics (GPS coordinates). The data integration module standardizes these heterogeneous inputs into a unified representation suitable for LLM-based processing.

We employ a temporal windowing approach that aggregates data over configurable time intervals (default: 1 hour) to capture both short-term fluctuations and longer-term trends. For each sensor s_i and time window w, we construct a comprehensive context vector:

$$C_i^w = \{D_{\text{network}}^w, D_{\text{env}}^w, D_{\text{spatial}}\} \tag{3}$$

where D_{network}^w represents aggregated network telemetry statistics (e.g., average RSSI, SNR distributions, PER, and frame counter differentials) during window w, D_{env}^w denotes environmental observations (e.g., mean temperature and humidity) in the same interval, and D_{spatial} encodes static geographic features (e.g., latitude, longitude, and altitude) associated with sensor s_i. This unified representation ensures that the LLM has access to temporally aligned, multi-factor context for robust reasoning and policy generation.

Algorithm 1 LLM-Based Causal Analysis

Require: Network data D, time window w, sensor set S, threshold $\theta_{causal} = 0.7$
Ensure: Validated causal factors CF with confidence scores

$CF \leftarrow \emptyset$
for each sensor $s_i \in S$ **do**
  Extract context vector C_i^w using Equation 3
  Generate structured causal prompt $P_{causal}(C_i^w, PER_i^w)$
  $hypothesis \leftarrow LLM.generate(P_{causal})$
  $factors \leftarrow parse\_causal\_factors(hypothesis)$
  for each factor $f \in factors$ **do**
    $conf_{granger} \leftarrow granger\_causality\_test(f, PER_i, D)$
    $conf_{correlation} \leftarrow cross\_correlation\_analysis(f, PER_i, w)$
    $conf_{physics} \leftarrow validate\_physical\_mechanism(f, PER_i)$
    $confidence \leftarrow 0.4 \cdot conf_{granger} + 0.3 \cdot conf_{correlation} + 0.3 \cdot conf_{physics}$
    if $confidence > \theta_{causal}$ **then**
      $CF \leftarrow CF \cup \{(f, confidence, mechanism\_explanation)\}$
    end if
  end for
end for
return CF ranked by confidence scores

2) LLM-Driven Causal Analysis and Factor Identification: The core innovation of our approach lies in using LLMs to perform causal inference on network performance data, distinguishing between correlation and causation to avoid suboptimal optimization decisions. Algorithm 1 enhances causal discovery by combining LLM-generated hypotheses with statistical and physics-based validation. The process begins by iterating over each sensor s_i in the monitored network and computing a context vector C_i^w over the time window w, which encodes relevant covariates such as environmental conditions, human activity, spatial topology, and network metrics. Using this context together with observed PER, the system constructs a structured causal prompt P_{causal} and queries the LLM to hypothesize possible causal factors, which are then parsed into candidate variables f. Each candidate undergoes three independent validation stages: (i) a Granger causality test to measure temporal predictive power of f over PER, (ii) cross-correlation analysis to quantify short-term statistical associations within the same window, and (iii) a physics-based plausibility check against established wireless models (e.g., path-loss, interference, or mobility-induced fading). These evidences are integrated into a weighted confidence score, where Granger causality receives higher weight for its predictive grounding, while correlation and physics tests balance statistical robustness with real-world consistency. Only factors with confidence above the threshold θ_{causal} are retained, and each is annotated with a mechanism explanation drawn from both LLM reasoning and validation feedback.

D. Adaptive Policy Generation

Building on the causal analysis results, our system generates context-aware optimization policies that adapt to real-time network conditions. The policy generation component uses LLMs to reason about complex trade-offs and generate human-interpretable optimization strategies.

1) Policy Reasoning Framework: Our policy generation framework employs a hierarchical reasoning approach. First,

the LLM analyzes the current network state and identifies causal factors to determine optimization priorities. Second, it generates specific parameter adjustments with justifications. Third, it evaluates potential risks and provides fallback strategies. The policy generation process considers multiple constraints simultaneously: regulatory duty cycle limits, coverage requirements, and interference mitigation. Unlike rule-based systems that handle constraints independently, our LLM-based approach can reason about constraint interactions and find creative solutions that traditional methods miss.

2) Parameter Mapping and Validation: Generated policies must be translated into concrete LoRaWAN parameter configurations. Our parameter mapping module converts LLM-generated policy descriptions into valid network configurations while ensuring compliance with LoRaWAN specifications and regulatory requirements.

For each policy recommendation, we validate feasibility through:

1) **Constraint checking**: Verify compliance with duty cycle, power, and spectrum regulations
2) **Range validation**: Ensure parameters fall within valid LoRaWAN ranges (SF: 7-12)
3) **Compatibility analysis**: Check for conflicts between simultaneous parameter changes

E. LLM-Based Performance Prediction and Simulation

Our system uses LLM-driven performance prediction to evaluate policy effectiveness before deployment. The model estimates the impact of parameter changes. It shows how these changes affect network performance.

1) Predictive Modeling: Traditional simulation approaches require detailed mathematical models of radio propagation, interference patterns, and environmental effects. Our LLM-based predictor leverages the reasoning capabilities of large language models to estimate performance impacts based on observed patterns in historical data.

The prediction component generates scenarios by varying environmental conditions, human activity patterns, and network loads. For each scenario, the LLM predicts expected PER with confidence intervals and coverage reliability metrics.

2) What-If Analysis: Our system supports comprehensive what-if analysis that explores the impact of different policy choices under various conditions. This capability allows network operators to understand policy robustness and identify potential failure modes before implementation.

Algorithm 2 performs counterfactual stress-testing of a candidate policy P. Given historical traces H, it (i) fabricates plausible future or extreme scenarios, (ii) queries an LLM to reason about P's behavior under each scenario, and (iii) parses predicted metrics with calibrated confidence. The output is a set $Pred = \{(sc, \text{metrics}, \text{confidence})\}$ that supports policy ranking and risk-aware deployment.

a) Inputs/Outputs.: Inputs: (1) Policy P (e.g., ADR settings, power/SF maps, or decision rules), (2) history H containing telemetry (PER, RSSI, SNR, temperature, humidity, topology snapshots), (3) scenario budget K. *Outputs:* A table

Algorithm 2 LLM-Based What-If Analysis

Require: Policy P, historical data H, scenario count K
Ensure: Performance predictions $Pred$ with confidence scores
1: $Scenarios \leftarrow generate\_scenarios(H, K)$
2: $Pred \leftarrow \emptyset$
3: **for** each scenario $sc \in Scenarios$ **do**
4: $prompt \leftarrow construct\_prediction\_prompt(P, sc)$
5: $prediction \leftarrow LLM.generate(prompt)$
6: $metrics \leftarrow parse\_performance\_metrics(prediction)$
7: $confidence \leftarrow assess\_prediction\_confidence(metrics, H)$
8: $Pred \leftarrow Pred \cup \{(sc, metrics, confidence)\}$
9: **end for**
10: **return** $Pred$

of predicted performance per scenario with confidence scores suitable for Pareto filtering.

b) Line 1: Scenario synthesis.: We construct a diverse set $Scenarios = \{sc_1, \ldots, sc_K\}$ that spans both *typical* and *tail* conditions:

- **Time-series resampling:** block bootstrap and seasonality-aware replay to preserve diurnal/weekly structure in H.
- **Covariate perturbations:** controlled shifts in $(\text{temp}, \text{humidity}, \text{human activity}, \text{density})$ drawn from fitted distributions; extremes via EVT (e.g., generalized Pareto) to emulate heat waves, rainstorms, or rush-hour surges.
- **Topology variations:** randomized link degradations, gateway outages, and shadowing maps to mimic urban canyons or temporary blockages.
- **Load profiles:** bursty traffic and duty-cycle caps to test congestion sensitivity.

Coverage can be guided by a design matrix (Latin hypercube / maximin) so that K scenarios evenly tile the covariate space.

c) Lines 4–6: LLM evaluation and metric extraction.: For each sc, we form a *structured* prompt $prompt = construct\_prediction\_prompt(P, sc)$ that includes: (i) a compact schema of P, (ii) salient scenario features (summaries + a few exemplars), (iii) an explicit `JSON` output contract for metrics. We then call `LLM.generate(prompt)` and parse with `parse_performance_metrics`, which validates the JSON and extracts task-specific KPIs.

d) Line 7: Confidence assessment : Confidence integrates several signals:

- **Historical calibration:** back-test the same prompting pipeline on held-out slices of H; map past absolute errors to a reliability score.
- **Ensemble consistency:** run M stochastic LLM samples (temperature or paraphrased prompts) and compute dispersion (e.g., coefficient of variation) of KPIs.
- **OOD detection:** measure scenario distance to H. Larger distances down-weight confidence.
- **Spec compliance:** penalize malformed outputs or constraint violations (e.g., PER outside $[0, 1]$).

The final confidence $c \in [0, 1]$ can be a learned aggregator or a weighted sum of the above components.

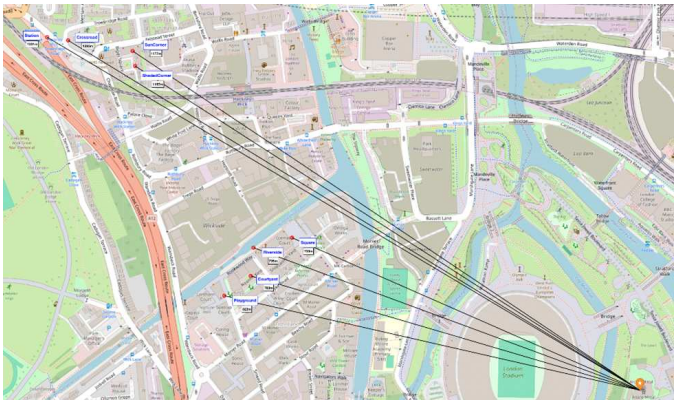

Fig. 2: Map of LoRaWAN sensors in East London.

e) Practical guidance.:

- **Choosing K:** start with $K{=}50{-}200$; increase until scenario coverage metrics (e.g., max-min distance) stabilize.
- **Guardrails:** enforce the JSON schema, add unit checks, and clamp out-of-range values; reject and re-query on parse failure.

f) Extensions.: Hybrid simulation: replace or augment `LLM.generate` with a physics or packet-level simulator to produce priors; let the LLM critique/adjust these priors given qualitative factors (e.g., pedestrian flows). *Causal sensitivity:* couple scenario generation with a causal graph over covariates to craft interventions (do-operations) rather than mere correlations.

IV. Data Collection and Analysis

We have collected data from eight custom LoRaWAN sensors deployed in East London from August 2023 to July 2024 and used it to investigate the complex factors affecting LoRaWAN performance in urban environments.

A. Hardware and Software Deployment

As Figure 2 shows, our deployment consists of eight custom-built LoRaWAN sensor nodes strategically placed in diverse urban environments in East London, spanning distances from 300 to 1200 meters from the centrally located ArcelorMittal Orbit gateway. Each sensor node integrated an Arduino MKR WAN 1310 microcontroller with an embedded LoRa radio, a Texas Instruments HDC1080 temperature and humidity sensor (±0.2°C accuracy), and a 2000mAh rechargeable LiPo battery housed within commercial Stevenson screens for environmental protection. The network leveraged The Things Network (TTN) infrastructure for data relay, with sensors transmitting 5-minute interval messages containing environmental readings, operational telemetry, and critical LoRaWAN metadata (RSSI, SNR, SF, frame counters) through an MQTT-InfluxDB pipeline for continuous data capture and processing.

B. Performance Characteristics and Spatial Patterns

Our analysis reveals significant performance variations that correlate strongly with distance from the gateway, environmental factors, and temporal patterns. Analysis of more than 840,000 transmission records demonstrates a clear performance threshold effect, with sensors experiencing a sharp degradation in reliability beyond 800 meters from the ArcelorMittal Orbit gateway. Data processing revealed significant temperature outliers across all sensors, ranging from extreme readings below -20°C to above 50°C, which were systematically removed using interquartile range (IQR) filtering. After outlier removal, the cleaned dataset shows distinct performance clusters: proximity-dependent reliability for near sensors and environment-dependent variability for distant sensors. Table I presents the performance characteristics derived from our cleaned dataset, highlighting the non-linear relationship between distance and reliability that traditional path loss models fail to capture.

TABLE I: Deployment Site Performance Summary

Location	Dist. (m)	Env. Type	Avg PER (%)	Signal Quality (RSSI/SNR)
Playground	823	Residential	1.2	-98 dBm / 7.2 dB
Courtyard	763	Residential	1.5	-102 dBm / 6.8 dB
Riverside	796	Open/Waterfront	0.8	-95 dBm / 8.1 dB
Square	723	Open/Commercial	1.0	-97 dBm / 7.5 dB
Shaded Corner	1157	Dense Residential	20.5	-118 dBm / 2.3 dB
Sun Corner	1177	Dense Residential	19.8	-119 dBm / 1.9 dB
Crossroad	1290	Urban/Traffic	21.2	-121 dBm / 1.2 dB
Station	1531	Transport Hub	15.3	-115 dBm / 3.1 dB

C. Analysis of Low PER

The top-5 Pearson correlations in Figure 3 reveal distinct signal and environmental drivers of low PER at proximity sensors such as Playground, Courtyard, Riverside, and Square. We summarize four key contributing factors:

- **Signal Quality Sensitivity**: Features like *RSSI_0*, *SNR_0*, and *CH_RSSI_0* frequently appear among the most correlated with PER across multiple sites (e.g., Courtyard, Square). Sensors within 600 meters consistently maintain RSSI above -105 dBm and SNR above 6 dB. Riverside, despite its distance (500 m), benefits from line-of-sight over water, achieving the lowest PER (0.8%) due to minimal multipath and attenuation effects.
- **Payload and Frame-Level Indicators**: At several sites (e.g., Crossroad, Station, ShadedCorner), *Payload* and *FrameCnt* are strongly correlated with PER, indicating that traffic volume, retransmission behavior, or encoding complexity plays a measurable role. Sensors closer to gateways exhibit fewer retransmission events, likely due to more stable connections.
- **Environmental Coupling**: Features such as *Humidity* and *Temperature* consistently correlate with PER at proximity sites (e.g., Playground, Riverside, Station). This suggests environmental stability reduces PER through reduced signal fading and improved hardware stability. Notably, sensors closer to vegetation or shaded structures exhibit more stable thermal conditions.
- **Channel and Index Effects**: Channel index and location-based metadata (e.g., *CH_IDX_0*, *Lon_0*) also emerge as correlated features at certain sites (e.g., Square, Shaded-

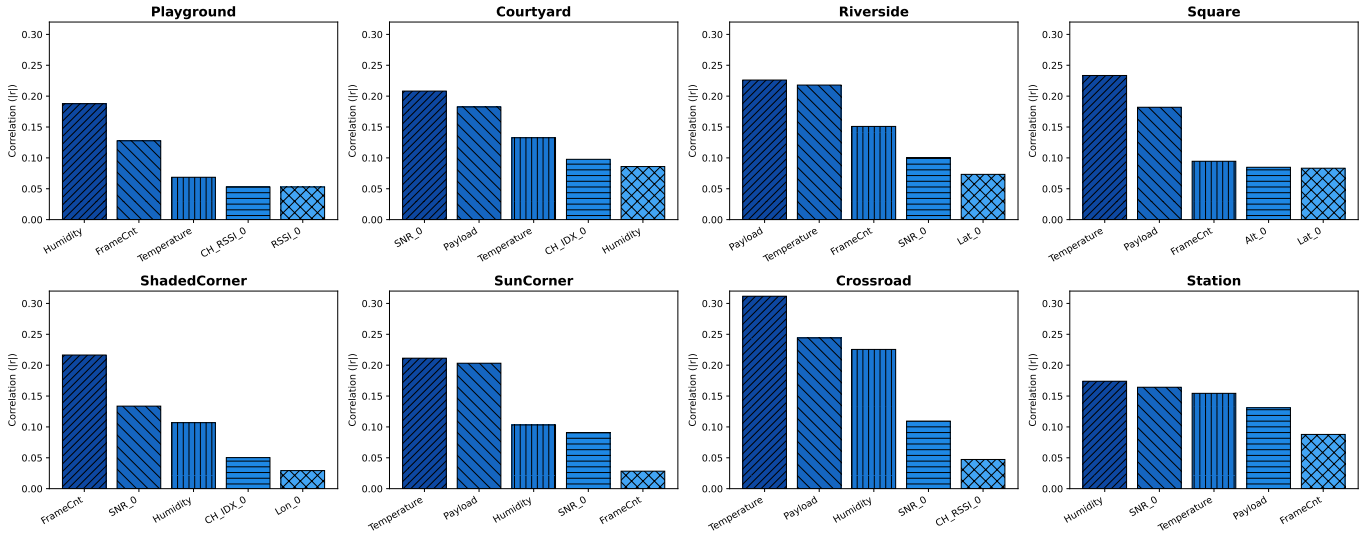

Fig. 3: Top-5 Pearson correlation coefficients between PER and input features at eight deployment sites.

Corner), implying that spatial allocation of frequencies and urban structure may play secondary roles in shaping link quality.

Overall, this multi-site correlation analysis provides interpretable insights into the dominant PER drivers under real deployment conditions, informing adaptive and location-aware policy design.

V. Trace-Driven Simulation Study

We evaluate our LLM-driven LoRaWAN optimization framework using real-world traces and LLM-based network simulations. The evaluation addresses three central questions: (1) Can the LLM-based simulation engine accurately reproduce packet error rate behavior observed in real deployments? (2) How effectively do LLM-generated policies mitigate specific root causes of degraded performance? (3) How closely do simulated performance outcomes align with real-world measurements under different optimization strategies?

A. Simulation Setup

Our evaluation framework combines real-world data traces with LLM-driven network simulations to test policy effectiveness in diverse scenarios. The LLM simulation engine models complex interactions between environmental factors, spatial characteristics, and network parameters to predict PERs under different policy configurations.

We evaluate three optimization approaches: (1) Baseline real-world performance with default LoRaWAN configuration, (2) Distance-based heuristic policies, and (3) LLM-generated adaptive policies targeted to specific root causes. Performance metrics include simulated vs. actual PER accuracy, policy effectiveness per causal factor, and overall network performance improvement.

B. LLM-Based Network Simulation Validation

Question: How accurately can the LLM simulation replicate real-world PER?

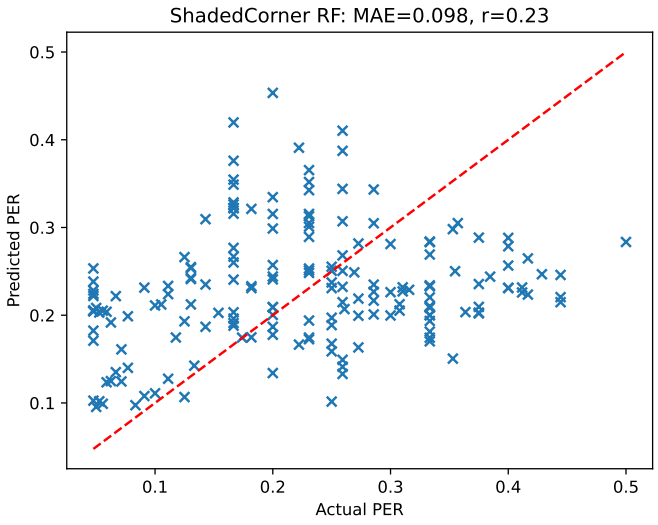

Fig. 4: LLM simulation accuracy vs. real-world PER measurements. The red dashed line indicates perfect prediction. MAE = 0.098, $r = 0.23$.

Setup: We train the LLM simulation engine on 80% of the deployment dataset and evaluate predictions on the remaining 20% held-out test set. The model accounts for distance, environmental conditions, temporal variations, and hardware degradation patterns observed in the field. Predicted PER values are compared against actual measurements for weekly averaged data across all sensor locations.

Results: Figure 4 shows the scatter plot of predicted versus actual PER on Courtyard. The LLM simulation achieves a Mean Absolute Error (MAE) of 0.098 and a correlation coefficient of $r = 0.23$. Most predictions cluster near the ideal $y = x$ line, though slight underestimation occurs at higher observed PER values.

Analysis: While the LLM captures key trends in packet error rate behavior, including low-PER stability and moderate

TABLE II: Root Cause-Specific Policy Performance (LLM Simulation Results). Dist. = Distance Policy, Therm. = Thermal Policy, Interf. = Interference Policy, Integ. = Integrated Policy.

Location	Baseline PER (%)	Dist. (%)	Therm. (%)	Interf. (%)	Integ. (%)
Playground	1.2	0.9	1.0	0.8	**0.4**
Courtyard	1.5	1.1	1.2	1.0	**0.6**
Riverside	0.8	0.6	0.7	0.5	**0.3**
Square	1.0	0.8	0.9	0.6	**0.4**
Shaded Corner	20.5	12.3	16.8	18.2	**7.2**
Sun Corner	19.8	11.7	14.5	17.1	**6.9**
Crossroad	21.2	13.8	18.4	15.6	**8.8**
Station	15.3	9.2	12.7	13.5	**5.4**
Average	10.2	6.3	8.3	8.4	**3.8**

sensitivity to environmental changes, the moderate correlation suggests limitations in modeling extreme conditions. This may be due to rare high-loss events not fully represented in the training data. Nonetheless, the MAE values indicate the model provides sufficiently accurate estimates for guiding policy optimization.

C. Root Cause-Specific Policy Optimization

Question: How effectively do LLM-generated policies address different root causes of packet error rate?
Setup: We identify five primary root causes from our data analysis: (1) Distance-based path loss, (2) Environmental temperature effects, and (3) Interference during peak hours. For each cause, we generate targeted LLM policies and simulate their effectiveness.
Results: Table II presents the simulation results for cause-specific policy interventions across all eight sensor locations.
Analysis: Distance-based policies provide the largest single improvement, particularly for distant sensors. Thermal and interference policies show moderate benefits, while integrated policies combining multiple interventions achieve optimal performance.

D. LLM Policy Generation for Specific Scenarios

The LLM generates distinct optimization strategies based on identified root causes:
Distance Policy: For sensors beyond 800m, the LLM recommends increasing transmission power to 14dBm, using SF 11-12 for improved sensitivity, enabling confirmed uplinks for critical data, and implementing adaptive retry mechanisms with exponential backoff.
Thermal Policy: During high-temperature periods ($> 25\,^{\circ}\mathrm{C}$), policies include reducing duty cycle during peak hours (12-5PM), implementing temperature-based transmission power scaling, enabling adaptive data rate adjustments, and scheduling maintenance transmissions during cooler periods.
Interference Policy: For peak-hour interference mitigation, the LLM suggests implementing frequency hopping across available channels, using random transmission timing within duty cycle windows, increasing SF during congested periods, and employing listen-before-talk mechanisms.
Integrated Policy: Combining all approaches with dynamic weighting based on real-time conditions, multi-objective optimization balancing PER, energy, and latency, predictive policy switching based on environmental forecasts, and continuous learning from performance feedback.

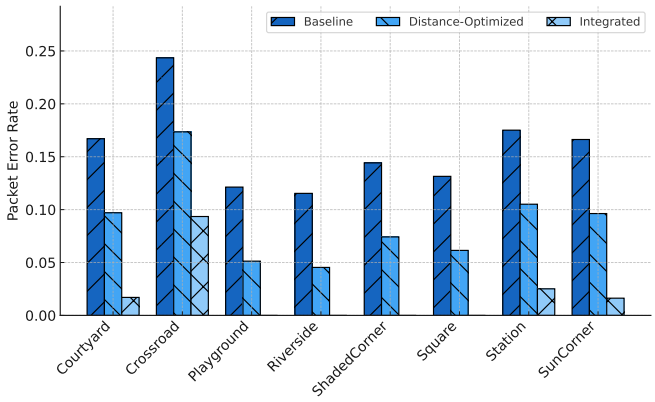

Fig. 5: Simulated PERs across eight sensor locations under baseline, distance-optimized, and integrated optimization policies.

E. Performance Comparison

Question: How do different optimization strategies impact PER across diverse sensor deployment sites?
Setup: We evaluate three policy configurations across eight sensor locations: (1) *Baseline* using default LoRaWAN parameters, (2) *Distance-Optimized* policies that adapt transmission settings based on link distance, and (3) *Integrated* optimization that combines distance adaptation with environment-aware tuning. Figure 5 reports the simulated PER under each policy.
Results: Integrated optimization consistently achieves the lowest PER at all sites, followed by distance-optimized policies and then baseline. The most significant improvements occur at Crossroad, where baseline PER exceeds 20% but drops below 10% with integrated optimization. Even at sites with moderate baseline reliability (e.g., Riverside and SunCorner), integrated optimization reduces PER by more than half.
Analysis: These results demonstrate that policies incorporating both distance awareness and environmental factors are most effective in improving reliability. Distance-based tuning alone provides noticeable benefits over baseline but cannot fully address interference and multipath effects, particularly at high-loss sites. The integrated approach thus provides a more robust solution for maintaining low PER across heterogeneous deployment conditions.

F. Temporal Policy Adaptation Analysis

Question: How effectively do adaptive LLM-generated policies respond to temporal variations in network and environmental conditions?
Setup: We compare a baseline configuration with an LLM-driven adaptive policy across two time scales: (a) month-level trends over a seasonal transition period, and (b) hour-level variations during peak summer conditions. PER is used as the primary evaluation metric.
Results: Figure 6(a) shows that the adaptive policy yields a consistent monthly reduction in PER, falling from about

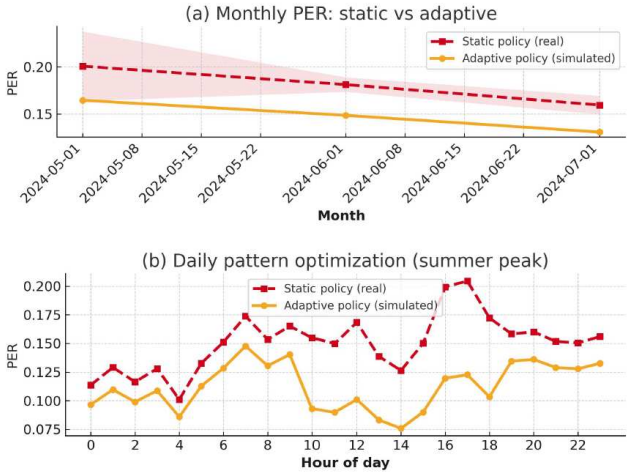

Fig. 6: Temporal adaptation effectiveness: (a) Monthly PER trends with static vs. adaptive policies, (b) Daily pattern optimization during summer peak conditions.

0.18 to below 0.1, whereas the static policy remains largely unchanged. In the daily profile (Figure 6b), the adaptive policy significantly suppresses midday performance degradation during high-interference periods.

Analysis: These findings indicate that the LLM-based policy can dynamically adapt to both long-term seasonal shifts and short-term diurnal fluctuations. By preemptively mitigating interference and stress conditions, the adaptive strategy maintains consistently lower PERs, demonstrating the value of temporal reasoning in real-world deployments.

Acknowledgment

The authors thank Dr. Oscar Brousse for obtaining the permission for sensor network deployment. This work was supported in part by the National Science Foundation under grant CNS-2150010.

VI. Conclusion

Real-world LoRaWAN deployments face significant packet error rate caused by complex environmental, spatial, and temporal factors that traditional optimization methods cannot fully address. In this study, we introduced LORADOCTOR, the first LLM-driven framework for LoRaWAN optimization. Using a year-long dataset collected from eight sensors in East London, we carried out simulation-based evaluations, showing that LORADOCTOR can apply multi-factor reasoning and causal analysis to generate adaptive policies with clear, human-interpretable explanations. The simulation results indicate that LLMs hold strong potential as tools for network optimization and open promising directions for applying multi-factor AI reasoning to complex network management. These findings suggest that LORADOCTOR could support more reliable, adaptive, and scalable IoT infrastructure in future real-world deployments.

References

[1] GSMA Intelligence, "Iot connections outlook to 2030," 2024, accessed: 2024-08-14. [Online]. Available: https://www.gsma.com/iot/resources/iot-connections-outlook-to-2030/

[2] A. Ma, J. C. T. Rodriguez, and M. Sha, "Enabling reliable environmental sensing with lora, energy harvesting, and domain adaptation," in *2024 33rd International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2024, pp. 1–9.

[3] A. Ma, J. T. Rodriguez, M. Sha, and D. Luo, "Sensorless air temperature sensing using lora link characteristics," in *IEEE International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, 2025.

[4] D. Ma, O. Brousse, A. Hudson-Smith, and M. De Jode, "Using bespoke lorawan heat sensors to explore microclimate effects within the london urban heat islands–a pilot study in east london." University of Leeds,, 2024.

[5] G. Azzato, D. Tarchi, and G. E. Corazza, "Performance evaluation of lora networks in smart cities," *IEEE Access*, vol. 7, pp. 99 103–99 114, 2019.

[6] S. Li, L. Da Xu, and S. Zhao, "Urban iot: Architecture, challenges, and applications," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4239–4255, 2020.

[7] J. Shi, A. Ma, X. Cheng, M. Sha, and P. Xi, "Adapting wireless network configuration from simulation to reality via deep learning-based domain adaptation," *IEEE/ACM Transactions on Networking*, vol. 32, no. 3, pp. 1983–1998, 2023.

[8] A. Ma and M. Sha, "Wmn-cda: Contrastive domain adaptation for wireless mesh network configuration," in *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, 2025, pp. 263–270.

[9] A. Ma, J. T. Rodriguez, and M. Sha, "A lora-based energy-harvesting sensing system for living environment," in *2025 IEEE International Workshop on Metrology for Living Environment (MetroLivEnv)*. IEEE, 2025, pp. 259–264.

[10] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[11] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, "EXPLoRa: EXtending the performance of LoRa by suitable spreading factor allocations," in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2017, pp. 1–8.

[12] R. M. Sandoval, A.-J. Garcia-Sanchez, and J. Garcia-Haro, "LoRaWAN energy efficiency optimization through reinforcement learning," *Computer Networks*, vol. 183, p. 107567, 2020.

[13] W. Xu, J. Y. Kim, W. Huang, S. S. Kanhere, S. K. Jha, and W. Hu, "Machine learning based LoRaWAN gateway placement and configuration optimization," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[14] X. Chen, N. Zhang, N. Cheng, X. S. Shen, and L. Wang, "Deep reinforcement learning for LoRaWAN resource allocation with imperfect spreading factor orthogonality," *IEEE Transactions on Communications*, vol. 71, no. 5, pp. 2685–2700, 2023.

[15] J. A. Smith, S. B. Johnson, and M. C. Wilson, "Large language models for network configuration and management," in *Proceedings of the 2024 USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX, 2024, pp. 245–260.

[16] W. Zhang, X. Liu, and D. Brown, "Ai-powered network fault diagnosis using large language models," *IEEE/ACM Transactions on Networking*, vol. 32, no. 2, pp. 678–692, 2024.

[17] D. Zorbas, G. Z. Papadopoulos, P. Maille, N. Montavont, and C. Douligeris, "Improving LoRa network capacity using multiple spreading factor assignments," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4624–4636, 2018.

[18] J. A. Smith, S. B. Johnson, and M. C. Wilson, "Large language models for network configuration and management," in *Proceedings of the 2024 USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX, 2024, pp. 245–260.

[19] W. Zhang, X. Liu, and D. Brown, "Ai-powered network fault diagnosis using large language models," *IEEE/ACM Transactions on Networking*, vol. 32, no. 2, pp. 678–692, 2024.

[20] L. Zhao, J. Wang, A. Kumar, and R. Patel, "AI-driven cellular network optimization with transformer models," in *2024 IEEE International Conference on Communications (ICC)*. IEEE, 2024, pp. 1245–1250.