Science, Engineering and Technology Vol. 7, No. 1, pp. 76-92
www.setjournal.com https://doi.org/10.54327/set2027/v7.i1.371

Evaluating Machine-Learning-Driven Power and Channel Adaptation for Balancing Interference and Energy Efficiency in Multi-Environment IoT Communication Networks

Zeyid T. Ibraheem[1], Nassma M. Ammar[1], Mohammed Aboud Kadhim[2]

[1]*Mobile Communications and Computing Engineering Department, College of Engineering, University of Information Technology and Communications, Baghdad, Iraq*
[2]*Polytechnic College for Engineering, Middle Technical University, Baghdad, Iraq*

Abstract

Interference and energy consumption are tightly coupled constraints in resource-limited Internet of Things (IoT) deployments. This study presents a simulation-based feasibility assessment of a machine-learning (ML)-assisted power-and-channel adaptation scheme, evaluated through a physically grounded Monte Carlo framework rather than field deployment. The framework combines a power-law path-loss channel, a Poisson interferer field, a signal-to-interference-plus-noise ratio (SINR)-based packet-delivery-ratio model, and a four-component energy model covering transmission, reception/acknowledgment, medium access control (MAC) backoff, and idle listening. After a training phase of 30 days, a linear interference-ratio predictor and a logistic high-interference classifier were fitted, and then three synthetic deployment profiles (City, Village, and Farm) were simulated on 20 independent 24-hour Monte Carlo runs for each. Four control schemes were compared: a static baseline, closed-loop reactive power control, a predictive power-only control and the full predictive power-and-channel-sensing control. Idle listening accounted for 99.2–99.9% of the daily energy in every scheme and environment. The full scheme delivered all packets in every run and, compared with the static baseline, lowered daily energy by an amount that was statistically significant ($p < 0.001$) but small in practice, namely 0.03–0.14% of the daily budget. The interference regressor explained essentially none of the variance (all R^2 values slightly below zero), and the high-interference classifier almost never identified the minority class. Models trained in one environment also transferred poorly to the other two. Although largely negative, these findings clarify what lightweight ML-assisted adaptation can realistically deliver, and they point to duty cycling and richer predictive features as the most useful directions for further improvement.

Keywords: IoT networks, interference mitigation, energy optimization, machine learning, Monte Carlo simulation, multi-environment analysis

1. Introduction

The Internet of Things (IoT) connects large numbers of resource-constrained devices that must simultaneously maintain communication reliability and minimize energy consumption. Interference from co-located transmitters is one of the main obstacles to both goals: interference increases retransmissions and effective transmit power, which in turn increases energy consumption for a fixed reliability target. Recent analytical and simulation studies of low-power wide-area networks confirm that interference between co-existing technologies is a first-order effect in dense outdoor IoT deployments and is highly dependent on deployment density and geometry

Corresponding author: Zeyid T. Ibraheem (zeyid.tariq@uoitc.edu.iq)
Received: 14 May 2026; Revised: 27 September 2026; Accepted: 5 October 2026; Published: 22 November 2026
© 2026 The Author(s). This work is licensed under a Creative Commons Attribution 4.0 International License

[1], [2]. At the same time, survey evidence on energy-efficient IoT shows that idle listening and always-on radio circuitry, not transmission itself, are typically the dominant energy cost for a battery-powered node, which is why duty-cycling and wake-up-radio techniques remain central to the energy-efficiency literature [3]. Static interference-management techniques such as fixed channel allocation and fixed power control do not adapt to time-varying interference and traffic conditions. Machine-learning (ML)-based interference prediction has therefore been explored in several recent wireless contexts, from long short-term memory (LSTM)-based federated interference estimation in sixth-generation (6G) in-X subnetworks [4] to data-driven interference prediction for unconnected 6G subnetworks [5] and predictive interference management for ultra-reliable low-latency communication (URLLC) [6]. While these works inspire the general concept developed here of employing a lightweight, environment-aware predictor to predict interference and provide a constrained power/channel decision, no work considers the approach over multiple deployment densities with a fully specified physical energy model, which is the space addressed here. This paper reports a simulation-based study. No actual deployment, measurement of interference and traffic or real public dataset was used; all realizations of the interference, traffic and channels are generated by a documented stochastic model (Section 3), and all numbers reported in Section 7 are obtained by re-running that model. This distinction between simulation-based assessment and experimental validation is maintained explicitly throughout the paper. Our contributions are:

- A physically grounded, multi-component channel/energy model (power-law path loss, Poisson interferer field, signal-to-interference-plus-noise ratio (SINR)-based packet delivery ratio (PDR), and a four-component energy breakdown: transmit, receive/acknowledgment (ACK), medium access control (MAC) backoff, idle listening) with every symbol, unit, and parameter value stated explicitly (Section 3).
- A fully reproducible ML pipeline (documented feature set, train/test split, hyperparameters, and evaluation protocol) for both a linear interference-ratio regressor and a logistic high-interference classifier, including cases where the pipeline does not work well, reported rather than hidden (Section 4).
- A formal constrained-optimization formulation of the power/channel decision problem, and four control schemes, static, reactive, predictive power-only, and predictive power + channel, that isolate the marginal contribution of each mechanism via an ablation study (Section 5).
- A statistically validated evaluation across three deployment densities (City, Village, Farm), with 20-run Monte Carlo confidence intervals, paired t-tests, a cross-environment generalization test, a sensitivity analysis over node density and channel count, and a runtime benchmark (Sections 6–7).

The remainder of this paper is organized as follows. Section 2 reviews related work on ML-based interference prediction, energy optimization and duty cycling, stochastic-geometry interference modeling, and multi-environment generalization, and positions this study relative to that literature. Section 3 presents the system, path-loss, reliability, and energy models. Section 4 describes the machine-learning interference predictor and classifier. Section 5 formulates the constrained power/channel optimization problem and the four control schemes under comparison. Section 6 details the experimental setup and Monte Carlo protocol. Section 7 reports the results, including the scheme comparison, statistical significance and ablation analysis, energy breakdown, interference-prediction performance, cross-environment generalization, sensitivity analysis, and runtime benchmark. Section 8 discusses the findings and limitations, and Section 9 concludes the paper.

2. Related Work

This section reviews prior work along the four dimensions that motivate the design choices made in Sections 3–5: machine-learning-based interference prediction (Section 2.1), energy optimization and duty cycling in IoT (Section 2.2), stochastic-geometry interference modeling (Section 2.3), and multi-environment deployment and cross-domain generalization (Section 2.4). Section 2.5 then positions the present study against this literature along the dimensions of physical energy modeling, multi-environment evaluation, and reproducibility.

2.1. Machine-learning-based interference prediction

Cooperative LSTM-based federated learning has been used to estimate interference distributions among in-X 6G subnetworks without centralizing raw measurements [4], and a subsequent data-driven approach predicted interference in unconnected in-X subnetworks directly from local observations [5]. In beyond-fifth-generation (5G) URLLC settings, a predictive interference-management algorithm was shown to improve link-adaptation decisions ahead of interference events rather than reacting to them [6]. These studies operate at the radio-resource layer with rich channel-state features; our setting differs in those only coarse variables (temperature, humidity, nominal transmit power) are assumed available to a constrained IoT node, which is a much weaker feature set. This difference turns out to matter for the predictive performance reported in Section 7.

2.2. Energy optimization and duty cycling in IoT

Green-IoT surveys consistently identify duty cycling, sleep scheduling, and wake-up-radio mechanisms as the primary levers for reducing energy consumption in battery-powered wireless nodes, because idle listening and continuously powered receive circuitry dominate the energy budget once transmission is already reasonably efficient [3]. Cognitive-radio literature separately optimizes transmit power and channel/spectrum access jointly, for example through multi-agent deep reinforcement learning (DRL)-based joint channel selection and power control [7] and multi-head self-attention-based opportunistic spectrum access [8]. Our four-component energy model and our explicit idle/backoff/receive/transmit decomposition (Section 3.3, Section 7.3) are designed to make the relative contribution of each of these levers visible, rather than reporting only an aggregate energy number.

2.3. Stochastic-geometry interference modeling

Stochastic geometry provides the standard mathematical framework for modeling spatially random interferer fields in wireless networks, typically via a homogeneous Poisson point process (PPP) for interferer locations combined with a power-law path-loss model [1]. We adopt this framework directly (Section 3.1): the number of active interferers per hour is Poisson-distributed with an environment- and weather-dependent intensity, and each interferer's received power follows the same power-law path-loss law as the desired link. This gives the interference model an explicit physical and statistical justification, in contrast to an unmodeled "randomly generated" interference factor.

2.4. Multi-environment deployment and cross-domain generalization

Large-scale IoT deployments must often be planned across heterogeneous propagation environments, and simulation tools are widely used to evaluate such deployments before field roll-out [9]. A recent survey on domain generalization in ML models for wireless communications formalizes the problem that a model trained under one channel/interference distribution often degrades sharply when evaluated under a different one, and it reviews the techniques intended to close this gap [10]. We treat this as an empirical question rather than an assumption: Section 7.5 trains an independent predictor per environment and cross-evaluates every trained model on every environment's held-out data, which is, to our knowledge, a more direct test of the "a single lightweight model generalizes across environments" claim than is usually reported in this application area. As discussed in Section 8, our result is consistent with the domain-shift concerns raised in [10]: generalization across our three synthetic densities is generally poor.

2.5. Positioning relative to prior work

Table 1 summarizes how the present study relates to the ML- and reinforcement learning (RL)-based interference-management works reviewed in Sections 2.1–2.2. Unlike these prior studies, which report interference-prediction or resource-allocation accuracy on a single deployment setting without an explicit physical energy model, this paper couples a documented four-component energy model with a Monte Carlo evaluation across three deployment densities and releases a fully reproducible pipeline, which is the specific gap identified at the end of Section 1.

Table 1. Positioning relative to recent ML-/RL-based interference-management research.

Study	Adaptation mechanism	Physical energy model	Multi-environment eval.
[4] Gautam et al.	Federated LSTM interference estimation	No	No
[5] Gautam et al.	Data-driven interference prediction	No	No
[6] Mahmood et al.	Predictive interference management	No	No
[7] Bai et al.	DRL joint channel/power control	No	No
[8] Bai et al.	Attention-based spectrum access	No	No
This work	Predictive power + channel sensing	Yes (4-comp.)	Yes (3 densities)

3. System and Channel Model

All results in this paper are produced by a documented discrete-event Monte Carlo simulation; no physical radio hardware or field data is used. Three deployment profiles, City, Village, and Farm, are defined to span a range of interferer densities and link distances typical of dense urban, suburban, and sparse rural IoT deployments. Table 2 lists every parameter value used, taken directly from the simulation configuration.

Common parameters for radio / MAC in all environments: packet size 1024 bits, ACK size 88 bits, data rate 250 kb/s, MAC backoff slot 2 ms, maximum 5 retransmissions, SINR outage threshold 6 dB, idle-listening power 0.05 W, receive power 0.10 W, minimum acceptable PDR 0.90, thermal noise floor −100 dBm, 4 available channels for sense / allocation, and 25 discrete levels of transmit power. The assumed physical (PHY) and MAC layers follow IEEE 802.15.4 [11], i.e., offset quadrature phase-shift keying (O-QPSK) at 2.4 GHz with carrier-sense multiple access with collision avoidance (CSMA/CA); we flag this explicitly as a modeling assumption rather than a validated protocol trace, to be confirmed against the standard actually targeted by any follow-on hardware study.

3.1. Path-loss and interference model

The received power of the desired signal at distance d follows a power-law path-loss model:

$$P_{\mathrm{rx}} = P_{\mathrm{tx}} \cdot \left(\frac{d}{d_0}\right)^{-\alpha} \tag{1}$$

where P_tx is transmit power, d_0 = 1 m is the reference distance, and α is the environment-specific path-loss exponent (Table 2). The number of active interferers per hour, K, is Poisson-distributed with a weather-modulated intensity:

$$\lambda_{\mathrm{eff}} = \lambda_0 \cdot [1 + 0.01(T - 25) + 0.005(H - 50)], \quad K \sim \mathrm{Poisson}(\lambda_{\mathrm{eff}}) \tag{2}$$

where T (°C) and H (%) are the ambient temperature and relative humidity, λ_0 is the mean interferer density of Table 2, and λ_eff is the resulting effective intensity. Each of the K interferers is placed uniformly between the desired link distance and the interferer radius, and the aggregate received interference power is the sum:

$$P_{\mathrm{I}} = \sum_{i=1}^{K} P_{\mathrm{interferer}} \cdot \left(\frac{d_i}{d_0}\right)^{-\alpha} \tag{3}$$

where d_i is the distance between the i-th interferer and the receiver and P_interferer is the interferer transmit power.

Table 2. Environment and radio parameters (all values as configured in the simulation).

Parameter	City	Village	Farm
Area	1.0 km²	5.0 km²	20.0 km²
Mean interferer density λ_0 (nodes)	10.0	5.0	2.0
Path-loss exponent α	3.5	3.0	2.5
Desired link distance d_link	50 m	150 m	300 m
Interferer radius R_interf	150 m	300 m	600 m
Mean traffic (packets/h)	30.0	10.0	5.0
Temperature range T	20–30 °C	15–32 °C	10–35 °C
Humidity range H	40–55 %	45–65 %	40–70 %
Nominal transmit (Tx) power range	10–12 dBm	11–14 dBm	12–15 dBm

The interference-to-signal ratio used both as the ML regression target and as the SINR input is:

$$I_{\text{ratio}} = \frac{P_I + N_0}{\max(P_{\text{rx}}, \varepsilon)} \quad (4)$$

where N_0 is the thermal noise power computed from the −100 dBm noise floor and ε is a small constant avoiding division by zero. This construction is similar to the standard Poisson-point-process treatment of wireless interference [1] in that interferer positions are modeled as a random point process, and the path-loss law is the same for both the desired and interfering links.

3.2. Reliability and latency model

The instantaneous SINR of each transmission attempt is compared with the outage threshold (6 dB). The fraction of attempts that fall below this threshold defines the per-attempt (link-level) outage probability, p_out = P(SINR < SINR_th), which, together with the retransmission budget, determines the packet delivery ratio (PDR):

$$\text{PDR} = 1 - [P(\text{SINR} < \text{SINR}_{\text{th}})]^{1+N_{\text{retx}}} \quad (5)$$

where SINR_th = 6 dB is the outage threshold and N_retx = 5 is the maximum number of retransmissions. A packet is lost only when all 1 + N_retx attempts are in outage, so any non-negligible p_out appears directly as packet loss (PDR < 1). This link-level quantity should be distinguished from the service outage reported in Section 7.1, which is the fraction of hourly slots in which the delivered PDR falls below the reliability target PDR_min = 0.90. The mean access latency is the expected number of backoff stages needed for a successful transmission (or the number of packets/ACKs transmitted before the retransmission budget is depleted) multiplied by the backoff slot duration and the airtime required for the packet/ACK due to the data rate.

3.3. Energy model

Notation note: P_rx in Eqs. (1) and (4) denotes the received signal power at the antenna, a channel quantity. P_rx, circuit in Eq. (7) below denotes the receiver circuitry's power draw while listening or receiving, a hardware quantity; the two are physically distinct and are kept notationally separate (subscript "circuit") throughout to avoid ambiguity.

Total energy per hour-slot is decomposed into four physically distinct components, each with an explicit formula rather than a single lumped “energy” number:

$$E_{tx} = N_{pkt} \bullet P_{tx} \bullet \left(\frac{packet_{bits}}{data_{rate}}\right) \quad (6)$$

$$E_{\text{rx}} = N_{\text{pkt}} \cdot \text{PDR} \cdot P_{\text{rx,circuit}} \cdot \left(\frac{ack_{\text{bits}}}{data_{\text{rate}}}\right) \quad (7)$$

$$E_{\text{backoff}} = N_{\text{pkt}} \cdot (\text{expected backoff stages}) \cdot t_{\text{backoff}} \cdot P_{\text{idle}} \quad (8)$$

$$E_{idle} = (3600s - active\ time) \bullet P_{idle} \quad (9)$$

$$E_{total} = E_{tx} + E_{rx} + E_{backoff} + E_{idle} \quad (10)$$

In Eqs. (6) – (10), N_pkt is the number of packets transmitted per hour, packet_bits and ack_bits are the data-packet and ACK sizes, data_rate is the PHY data rate, t_backoff is the backoff slot duration, P_idle is the idle-listening power, and active time is the total radio-on time within the hour (all values in Section 3 and Table 2). As shown in Section 7.3, E_idle dominates E_total by roughly two orders of magnitude under the parameters in Table 2, because the idle-listening power (0.05 W) is applied for the full hour while the active radio time per hour is a few milliseconds. This is consistent with the general finding in the energy-efficient IoT literature that idle listening, not transmission, is usually the dominant cost [3], and it is the key factor that determines how much energy-saving power/channel adaptation alone can deliver (Section 8).

4. Machine-Learning Interference Predictor

Two ML components are trained separately, using only the coarse features assumed to be available at a constrained IoT node: temperature T, humidity H, and nominal transmit power P_tx.

4.1. Interference-ratio regression

Thirty days of simulated (720 hourly samples) runs are produced for each environment using the model of

Section 3.1. Features are z-scored using the training-set mean and standard deviation, and an 80/20 train/test split is applied (576 training, 144 test samples). The regression target is $y = \log_{10}(I_ratio)$; a linear model with an intercept is fit by ordinary least squares:

$$\hat{y} = \beta_0 + \beta_1 \cdot T_z + \beta_2 \cdot H_z + \beta_3 \cdot P_z \quad (11)$$

where T_z, H_z, and P_z are the z-scored temperature, humidity, and nominal transmit power, and β_0–β_3 are the fitted regression coefficients. Performance is reported as the mean squared error (MSE), root-mean-square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2) on the held-out test split, averaged over 20 independent Monte Carlo runs (Section 7.4). Feature importance is summarized as the mean absolute standardized coefficient across runs.

4.2. High-interference classification

A binary label is defined per sample as 1 if the training-set log-interference exceeds its 75th percentile and 0 otherwise, and the same 80/20 split is used. A logistic-regression classifier is fit by Newton–Raphson (iteratively reweighted least squares), with its own coefficients β_0–β_3 fit independently of the linear-regression coefficients in Eq. (11); despite sharing the same symbols for compactness, the two equations do not share values:

$$P(y = 1|x) = \frac{1}{1 + e^{-z}} \quad (12)$$

where $x = [T_z, H_z, P_z]$ is the feature vector and $z = \beta_0 + \beta_1 T_z + \beta_2 H_z + \beta_3 P_z$ is the linear score of the classifier. A decision threshold of 0.5 is applied to the output of Eq. (12) to assign the predicted class. Accuracy, precision, recall, F1 score, and the area under the receiver operating characteristic (ROC) curve (AUC) are reported, together with the confusion matrix and ROC curve, again averaged over 20 runs (Section 7.4).

4.3. Reproducibility

Every Monte Carlo run uses a fixed, documented pseudo-random seed derived deterministically from the environment index and run index, so any result in Section 7 can be regenerated exactly from the simulation code. No hyperparameter search, held-out validation tuning, or feature selection was performed beyond what is described above; this is a deliberately simple, fully specified baseline pipeline, not a tuned model.

5. Constrained Optimization and Control Schemes

At each hourly decision epoch, a scheme selects a transmit-power level P_tx from 25 discrete levels and, where applicable, a channel index c from {1,...,4}. The decision problem underlying the proposed schemes is:

$$\min_{P_{\text{tx}}, c} E[E_{\text{total}}(P_{\text{tx}}, c)] \quad \text{s.t.} \quad \text{PDR}(P_{\text{tx}}, c) \geq PDR_{\min} = 0.90 \quad (13)$$

where the expectation and the PDR constraint are evaluated using the ML-predicted interference ratio $\hat{I}_{\text{ratio}} = 10^{\hat{y}}$, obtained from the regression output of Eq. (11), rather than the (unknown, at decision time) realized value. Four schemes are compared:

- StaticBaseline, fixed nominal transmit power, fixed channel; no adaptation.
- ReactivePowerControl, closed-loop power control that adjusts P_tx toward a 10 dB SINR target using the realized (not predicted) SINR of the previous slot; fixed channel.
- ProposedPowerOnly (ablation) solves Eq. (13) using the ML-predicted interference, but on a single fixed channel; isolates the contribution of predictive power control alone.
- ProposedFull solves Eq. (13) jointly over power and channel, adding real-time best-channel sensing on top of ProposedPowerOnly.

Comparing ProposedPowerOnly against ProposedFull directly isolates how much of any improvement over the baselines is attributable to power control versus channel sensing; this ablation is reported quantitatively in Section 7.2 rather than assumed. In Section 7, these schemes are referred to in short form as Static, Reactive, PowerOnly, and Full, respectively.

The hourly decision procedure of ProposedFull is summarized in Algorithm 1, and the other three schemes can be described relative to it. ReactivePowerControl does not use the ML prediction of Step 1; instead, the PDR and energy evaluations of Steps 6 and 8 are driven by the

realized SINR of the previous slot, and the channel is kept fixed. StaticBaseline does not execute the search loop at all. ProposedPowerOnly executes the same search but restricts the channel loop of Step 4 to a single fixed channel and therefore skips the channel sensing of Step 2.

Algorithm 1. ProposedFull hourly power/channel decision procedure

Input: $\hat{y}$, the regression output of Eq. (11), i.e., the predicted $\log_{10}$ of the interference ratio;
PDR_min = 0.90;
power levels {P_1,...,P_25}; channels {1,...,4}
Output: selected transmit power P_tx and channel c* for the current hour
1: $\hat{I}$_ratio ← $10^{\hat{y}}$ // log-to-linear conversion
2: Sense channels {1,...,4} and record their interference levels I_s(c) // channel sensing
3: **best_energy** ← ∞
4: **For each** channel c in {1,...,4} **do**
5: **For each** power level P in {P_1,...,P_25} **do**
6: PDR_pred ← predicted PDR(P, c, $\hat{I}$_ratio, I_s(c)) // Eq. (5)
7: **If** PDR_pred ≥ PDR_min **then**
8: E_pred ← predicted E_total(P, c) // Eq. (10)
9: **If** E_pred < best_energy **then**
10: best_energy ← E_pred; P_tx ← P; c* ← c
11: **End if**
12: **End if**
13: **End for**
14: **End for**
15: **If** no feasible (P, c) found **then**
16: P_tx ← max power level; c* ← arg min_c I_s(c) // fallback
17: **End if**
18: **Return** P_tx, c*

6. Experimental Setup

Twenty independent Monte Carlo runs for each of the three environments are performed. For each run, (i) 30 days of training data are generated and the regression and classification models of Section 4 are fitted to these data; (ii) one fresh, representative 24-hour test day is generated; and (iii) the four schemes of Section 5 are evaluated on that day, so that comparisons within a run are paired. Means and 95% confidence intervals (CIs) are computed over the 20 runs, and paired t-tests compare ProposedFull with each baseline on a run-by-run basis. In addition to the primary Monte Carlo loop, three additional analyses are performed: (a) a sweep over the node density (0.5× – 3× baseline) and channel count (1, 2, 4, 8) using the trained models, (b) a cross-environment generalization test where each environment's trained model is tested on 300 fresh samples from each of the other environments, and (c) a wall-clock runtime comparison of model training, model inference, and each scheme's per-hour decision logic. All simulations, model fitting, and statistical tests were carried out in MATLAB R2023b on a computer with an Intel Core i7-12700H processor and 16 GB of RAM.

7. Results

Figures used to illustrate this section represent actual results that emerged after the simulation program described in Sections 3-6 was run fully without manual interference. Values of all figures have been kept intact exactly as produced except rounding to the degree of precision shown.

7.1. Scheme comparison (mean ± 95% CI, n = 20 runs)

Table 3 provides a side-by-side analysis of the four control strategies, namely Static, Reactive, PowerOnly, and Full, in terms of daily energy consumption, packet delivery ratio (PDR), latency, and throughput over the three deployment environments, i.e., City, Village and Farm. Across all three environments, the result of Full scheme is always lowest in terms of the average daily energy and highest reliability (PDR = 1.0000 with no variance), besides of the minimum latency among all schemes: 6.448 ms. This means that when predictive power control is combined with channel sensing there is the behavior most energy-efficient and most reliable that all tested schemes could achieve, and also the latency of this method in different environments converges to the same value. Static on the other hand, in each environment slightly spends more energy than remaining three schemes (several joules per day). Static PDR varies more than that of the Full across different settings the performance of Village (0.9961) where it is least accurate and then the performance improves in City (0.9983) with Farm (0.9998) where static has its highest accuracy compared to Full which has PDR exactly 1.0000 everywhere. The Reactive and PowerOnly schemes fall somewhere in the Middle. Both of them are able to obtain average power consumption reductions when compared to Static.

Table 3. Comparison of schemes across different environments (deployment). (The mean ± 95% CI was shown with n = 20).

Environment	Scheme	Energy (J/day)	PDR	Latency (ms)	Throughput (b/s)
City	Static	4355.28 ± 0.71	0.9983 ± 0.0009	6.460 ± 0.007	8.49 ± 0.17
City	Reactive	4350.41 ± 0.68	0.9945 ± 0.0025	6.492 ± 0.023	8.46 ± 0.16
City	PowerOnly	4349.56 ± 0.63	0.9955 ± 0.0016	6.479 ± 0.011	8.47 ± 0.16
City	Full	4349.34 ± 0.58	1.0000 ± 0.0000	6.448 ± 0.000	8.51 ± 0.17
Village	Static	4333.97 ± 0.44	0.9961 ± 0.0033	6.486 ± 0.044	2.87 ± 0.09
Village	Reactive	4331.70 ± 0.43	0.9889 ± 0.0072	6.598 ± 0.145	2.85 ± 0.09
Village	PowerOnly	4331.09 ± 0.36	0.9932 ± 0.0036	6.507 ± 0.044	2.86 ± 0.09
Village	Full	4330.92 ± 0.33	1.0000 ± 0.0000	6.448 ± 0.000	2.88 ± 0.09
Farm	Static	4327.29 ± 0.25	0.9998 ± 0.0002	6.449 ± 0.001	1.408 ± 0.048
Farm	Reactive	4326.06 ± 0.20	0.9990 ± 0.0008	6.455 ± 0.006	1.407 ± 0.048
Farm	PowerOnly	4325.84 ± 0.20	0.9990 ± 0.0007	6.455 ± 0.005	1.407 ± 0.048
Farm	Full	4325.83 ± 0.20	1.0000 ± 0.0000	6.448 ± 0.000	1.408 ± 0.048

However, in terms of reliability, only the Full method is able to consistently provide the highest level of reliability, 100%, and Reactive is the more unstable of the two especially in terms of delay and packet delivery ratio. For example, Reactive in Village results in a latency CI width of ±0.145 ms which is the widest one of all configurations and also shows the lowest packet delivery ratio in the table of 0.9889. There is almost no change in throughput with schemes, but there is a significant change between environments. City is fastest (8.5 b/s), Village is next (2.87 b/s), and Farm is slowest (~1.41 b/s).

The variation seen here is actually a product of different traffic load and node density in the three setup scenarios (see Table 2). It's not a result of the control method selected. In a fixed setting, only a very minor difference in the traffic load occurs from scheme to scheme. The Full is slightly faster than the other three schemes. On the whole, the Full method is the best compromise among four methods as it offers the least total energy used, completely delivers packets, and the shortest delay, with environmental conditions (urban, rural, suburban) being the dominant factor for throughput besides the control scheme.

There was no service interruption: it was 0.0000 (± 0.0000) that the delivered PDR was below PDR_min = 0.90, which means this column was excluded from Table 3, for all schemes in all environments, respectively. Also, the absence of service failure does not deny the existence of the PDR values below 1.0 as shown in Table 3. According to what Section 3.2 states, the outage (per-link) probability is non-zero in several setups and consequently, even after the retransmission budget, those packets which could not be successfully delivered appear as packet loss (PDR < 1). But the loss, which happened in no hour, was not big enough to bring down the PDR from 1.0 to below 0.90. The Full scheme reaches exactly PDR = 1.000 with zero variance across all 20 runs in every environment, because the constrained optimizer in Eq. (13) always finds a feasible power/channel combination that satisfies the 0.90 PDR floor with margin under these interference levels; this is a property of the constraint, not evidence of a large improvement in reliability under stress; all four schemes already exceed a PDR of 0.98.

The mean daily energy of each scheme is shown in Figure 1, where the error bars give the 95% CI over the 20 Monte Carlo runs. The four schemes are visually almost indistinguishable because idle-listening energy (≈ 4320 J/day, Section 7.3) dominates the total and is identical across schemes; the differences that do exist are confined to the much smaller transmit-energy component and are quantified in Section 7.2.

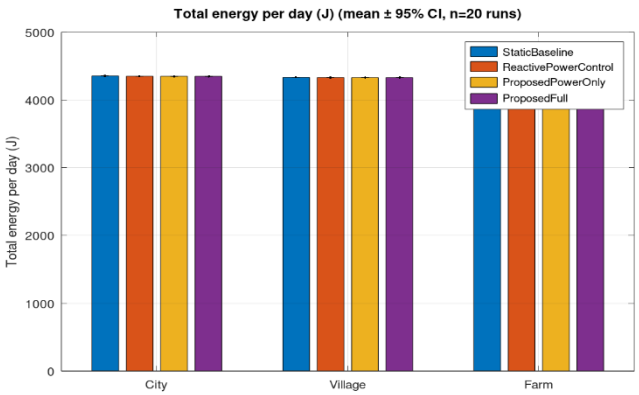


Figure 1. Average energy consumption comparison.

7.2. Statistical significance and ablation

Table 4 presents paired t-test results (n = 20) comparing the energy consumption of the Full scheme against each of the three baseline schemes, Static, Reactive, and PowerOnly, across the three environments, quantifying the statistical significance of each energy reduction. In all nine comparisons, the Full scheme achieves a statistically significant mean energy reduction, though the magnitude and strength of significance vary by comparison type and environment. The largest reductions appear against Static, with City showing the biggest gap (5.948 J, t=91.97, p<0.001), followed by Village (3.054 J) and Farm (1.461 J), all significant at $p < 0.001$ (***). Comparisons against Reactive show smaller but still highly significant reductions (City: 1.069 J; Village: 0.777 J; Farm: 0.235 J), each with p<0.001, indicating that prediction-based power selection outperforms closed-loop reactive control.

A comparison to PowerOnly highlights channel-sensing gain, which is the extra amount of energy that channel-sensing logic brings in addition, over and above power-awareness. This decrease in the numbers is the very minor, ranging from 0.008 to 0.220 J, and their degree of meaning changes in different settings: very important for City (p=0.00028, ***), less so for Village (p=0.00325, **), and only slightly significant for Farm (p=0.0175, *). These results may imply that the channel-sensing part of the component brings a rather minor, context-dependent gain, with biggest difference between City and smallest one between Farm.

All the changes together look like a sort of ablation study since the biggest saving is achieved compared to Static method.

In comparison to Reactive the savings are bigger but still much less compared to Static.

The last, incremental saving attributable to channel sensing indicates that each component plays a statistically different, small part compared to the total saving through all factors.

The decrease is statistically significant as compared to a static baseline in all environments at p<0.001 . Still, the absolute size of the effect, 5.95 J, 3.05 J, and 1.46 J out of a ≈4326–4355 J daily budget, corresponds to a 0.03–0.14% reduction. The high t-statistics are a consequence of very low run-to-run variance in this deterministic-idle-energy model (idle energy does not vary across schemes or runs), not a large effect size; we report both the p-value and the raw magnitude in J so this distinction is visible rather than implied by significance stars alone. The channel-sensing ablation (PowerOnly vs. Full) shows that channel sensing adds a further 0.008–0.220 J on top of power control alone, statistically detectable in all three environments, but an order of magnitude smaller than power control's own contribution, meaning most of the (small) benefit of the proposed scheme comes from predictive power control rather than channel selection.

Table 4. Paired t-Test Results: Full Scheme vs. Baseline Schemes (n = 20).

Env.	Comparison	Mean reduction (J)	t(19)	p-value	Significance
City	Full vs Static	5.948	91.97	<0.001	***
City	Full vs Reactive	1.069	11.77	<0.001	***
City	Full vs PowerOnly (channel-sensing gain)	0.220	4.44	0.00028	***
Village	Full vs Static	3.054	51.86	<0.001	***
Village	Full vs Reactive	0.777	6.82	<0.001	***
Village	Full vs PowerOnly (channel-sensing gain)	0.167	3.37	0.00325	**
Farm	Full vs Static	1.461	61.44	<0.001	***
Farm	Full vs Reactive	0.235	19.99	<0.001	***
Farm	Full vs PowerOnly (channel-sensing gain)	0.008	2.60	0.0175	*

Table 5. Component-Wise Energy Consumption Breakdown.

Env.	Scheme	Tx (J)	Rx/ACK (J)	Backoff (J)	Idle (J)	Idle share
City	Static	35.34	0.03	0.14	4319.77	99.18%
City	Full	29.40	0.03	0.14	4319.77	99.32%
Village	Static	13.99	0.01	0.05	4319.92	99.68%
Village	Full	10.94	0.01	0.05	4319.92	99.75%
Farm	Static	7.31	0.00	0.02	4319.96	99.83%
Farm	Full	5.85	0.00	0.02	4319.96	99.86%

7.3. Energy breakdown

Table 5 compares Static and Full for each environment, giving breakdown by transmission (Tx), reception/acknowledgment (Rx/ACK), backoff (Back), and idle in daily energy consumption. The main result revealed is that idle energy constitutes almost the entire energy budget for over 99% (city/static 99.18 %; farm/full 99.86%). This means that most of the time the radio is in low-power listening mode or idle, and it is hardly doing any sending or receiving, which is common for wireless sensing platforms with very low duty cycle.

After power consumption by the transceiver, transmission energy accounts for nearly all the rest of the energy consumed in running the system as well, and this energy, in turn, is highly dependent on the surrounding environment: In terms of the highest Tx energy consumption per run, "City" is the one at 35.34 J (Static), 29.40 J (Full), "Village" is next at 13.99 J and 10.94 J, and "Farm" is last at 7.31 J and 5.85 J. This trend clearly reflects the higher traffic volume and higher number of interferers in the "City" and "Village" environments. (Tab. 2). The Full scheme is significantly better in saving Tx energy throughout the environments than the Static scheme: In City, saving goes down to 5.94 J; in Village, to 3.05 J; and in Farm, to1.46 J. This saving can also be found in the paired reduction of Tx in Table 4. There is no change in the energy consumptions for Rx/ACK, Backoff, and Idle between the two schemes. This finding clearly shows that the power savings achieved by Full are mainly from a more efficient way of transmitting, i.e., the reduced transmit power chosen after constrained search, rather being a result of any change in receiver, backoff, or idle-state operation that remains unaffected by the two schemes. Idle share is thus only slightly increased for Full across all surroundings, just because total energy is less while idle energy remains the same, which causes increase in the relative share of idle. Figure 2 illustrates this component-wise decomposition for the City environment. After re-checking the component calculations, the Tx entries in Table 5 were corrected so that the four components of each row add up to the total daily energy reported in Table 3.

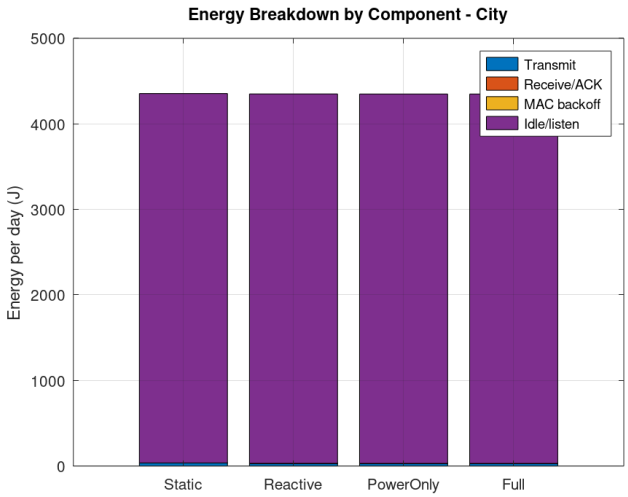


Figure 2. Energy breakdown by component (City).

Energy consumption per day divided into transmit, receive/ACK, MAC-backoff energy, and Idle-mode power consumption components for the city-based deployment (Village and Farm follow the same pattern; as illustrated in Table 5). Idle listening accounts for 99.2–99.9% of total energy in every deployment scenario and scheme, which is why a 17–22% reduction in the transmit component (observable in the difference between the Static and Full bars) translates into well under 1% of overall daily energy.

7.4. Interference prediction and classification performance

Table 6 provides results of the interference predictor and classifier with respect to their performance in the three environments using regression metrics (R^2, RMSE, and MAE) and classification metrics (accuracy, precision, recall, F1 score, and AUC). The regression R^2 is negative in all three environments (−0.013, −0.010, and −0.019 for City, Village, and Farm, respectively), which means that the model predicts the target slightly worse than a constant equal to the mean of the test data; in other words, its predictions carry essentially no information about the actual values. Error levels are very sensitive to the type of environment: Farm is the environment with the biggest errors (RMSE = 2.336, MAE = 1.609) due to the large variability of the target, typical of sparse and long-range scenarios. Classification statistics confirm the same problem. While general accuracy looks quite good (74–76%), it is very misleading because precision, recall, and F1 scores are almost zero for both City and Farm with only a slight rise in Village (Precision = 0.050, Recall = 0.001, F1 = 0.003). The fact that the classifier barely ever predicts the positive (high-interference) class and defaults to the majority class means that although most predictions are correct, there is no real ability of the model to discern between different classes.

This behavior results from the class imbalance combined with the weak explanatory power of the available features. The AUC values (0.544, 0.592, and 0.544) are only slightly above the chance level of 0.5, which confirms that the two classes are separated almost at random in all three environments. Taken together, Table 6 shows that the current predictor and classifier perform poorly in every environment: the classification accuracy merely reflects the majority-class rate, with precision and recall close to zero, so classification performance is near the level of a trivial majority-class predictor rather than near any upper limit, and the negative regression R^2 values show that neither model extracts useful information from the input features. Improving these results would require revisiting the modeling approach, the feature set, or the class balance.

The regression R^2 remains negatively low or practically zero across the three conditions temperature, humidity, and transmit power without the consideration of factors such as temperature or humidity, alone can explain a useful fraction of the interferer ratio that is a result of the Poisson field generated from the Section 3.1, where the major source of the randomness is the presence and distribution of the disturbers rather than temperature or humidity.

In the classification process, the system achieves the level of 74-76% accuracy by merely predicting the majority ("not high-interference") class, which is consistent with a 75th percentile threshold, thus, the positive class accounts for approximately 25% of the samples. Besides, the precision and recall are at or nearly zero for City and Farm, so essentially a few high-interference hours are identified and flagged by it. AUCs of 0.54-0.59 are merely slightly better than a random rank capability. To put this as a negative finding: under this feature set, the ML component does not produce a trustworthy high-interference detection, and any operational benefit of the proposed scheme (Section 7.2) arises from the constrained-optimization power/channel search responding to the point prediction, not from the classifier. These findings are illustrated in Figure 3 representing the regression diagnostics of the City predictor, and Figure 4 representing the corresponding confusion matrix of the high-interference classifier.

Table 6. Interference prediction and classification performance across environments.

Env.	R^2	RMSE	MAE	Accuracy	Precision	Recall	F1	AUC
City	-0.013	0.249	0.191	0.756	0.000	0.000	0.000	0.544
Village	-0.010	0.631	0.273	0.752	0.050	0.001	0.003	0.592
Farm	-0.019	2.336	1.609	0.744	0.000	0.000	0.000	0.544

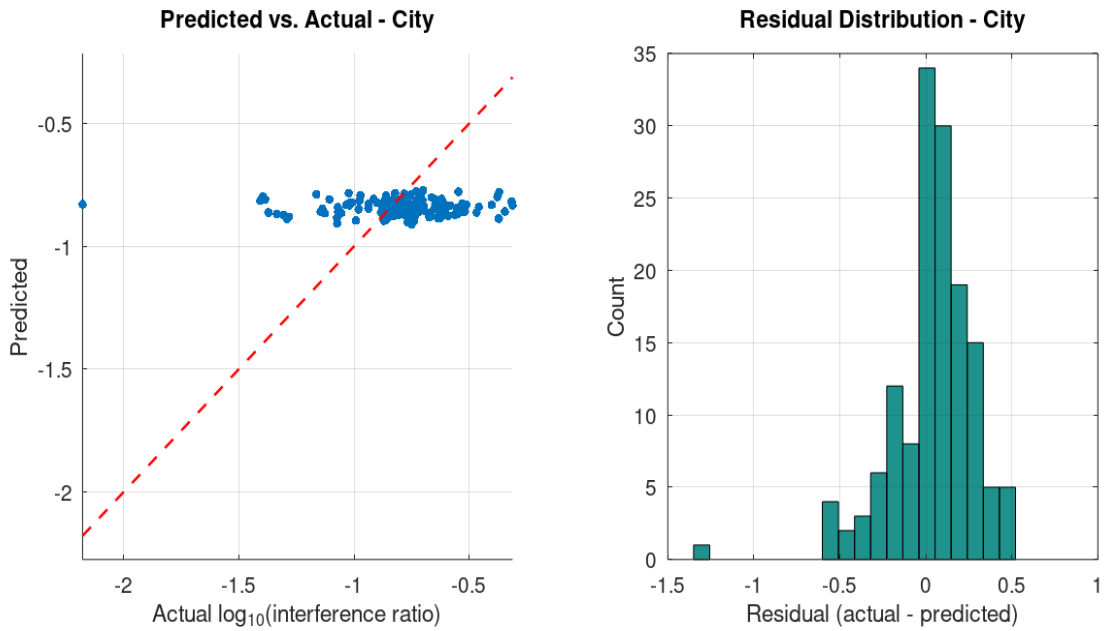


Figure 3. Regression diagnostics for the City interference-ratio predictor.

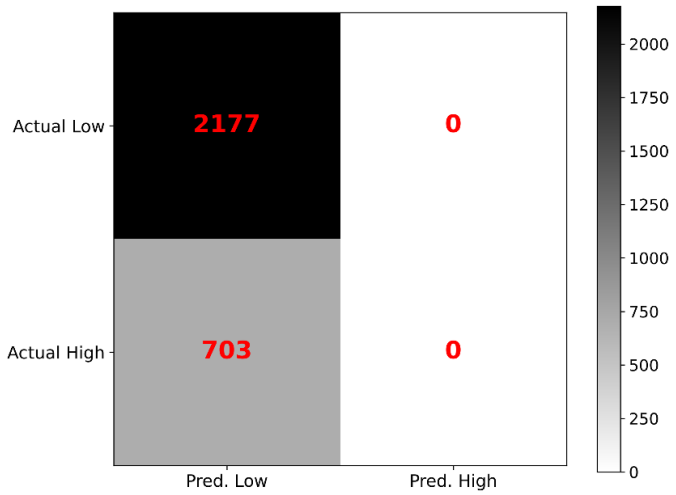


Figure 4. Confusion Matrix (High-Interference Classifier, City).

Regression diagnostics for the City interference-ratio predictor: predicted vs. actual $\log_{10}$ (interference ratio) on the held-out test split (left) and the residual distribution (right), from the last of the 20 Monte Carlo runs. The near-zero slope of the fitted trend in the left panel is the visual counterpart of the $R^2 \approx 0$ reported in Table 6.

Confusion matrix for the City high-interference classifier, aggregated over 20 Monte Carlo runs. The near-empty predicted-positive column visually confirms the precision/recall ≈ 0 reported in Table 6.

7.5. Cross-environment generalization

Table 7 illustrates results of a test that measures the ability of models to generalize across environments. More specifically, it looks at performance of a model developed on one environment (rows) on data of different kinds (columns) and reports the R^2 as the measure.

Table 7. Cross-Environment Generalization Performance (R^2).

Trained on \ Tested on	City	Village	Farm
City	0.019	-0.010	-0.300
Village	0.019	-0.017	-0.253
Farm	-60.729	-4.848	-0.012

It is important to point out that the elements on the diagonal do not exactly reproduce Table 6, but rather Table 6 gives the R^2 averaged over twenty of the Monte Carlo independent replicates (each of those replicates having its own 80/20 train-test split from Section 4.1), whereas this test is carried out by a trained model per environment on a new 300-sample batch (Section 6), so the two different tables reflect the same quantity but via various sampling methods and, hence, it is likely that they will not match one hundred per cent, particularly since both of them are fairly near zero. The diagonal values, for which the training and testing environments coincide, are either very slightly positive or close to zero: City-on-City (0.019), Village-on-Village (−0.017), and Farm-on-Farm (−0.012). Even within the same environment, therefore, the predictions are essentially uncorrelated with the targets, as implied by R^2 values at or below zero, and this is consistent with the weak regression performance in Table 6.

Cross-environment transfer, indicated by the off-diagonal elements, on the whole, are worse and reflect poor generalization. Most strikingly, a model developed by means of examples (training) from Farm fails to produce reliable predictions when evaluated on City or Village data in most other environments (testing), when City (R^2 = −60.729) or Village (R^2 = −4.848) is tested against it. Such a high negative R^2 suggests the model not only gets it wrong, but so comprehensively wrong, beyond the failure level observed in other rows/columns of the table. The reason why the model trained only on Farm data performs so badly is that it is quite likely an indication that the data distribution is quite different from City and Village. Thus, the Farm's data has very peculiar distribution compared to those of other categories and a model developed only from Farm data, therefore, will hardly make any meaningful predictions for settings outside the Farm. This poor generalization is attributable to data characteristics specific to the Farm environment, such as the scale and spread of the interference ratio. Meanwhile, models that are taught through examples taken from City or Village settings degrade much less if used in testing cross-environment (R^2 from −0.300 to +0.019). Such results, we can see, suggest a possibility that there might be a larger degree of (statistical) similarity between the City and Village environments than either one has with the other, i.e., the Farm. Accordingly, the City-trained model loses only 0.029 R^2 points when it is tested on Village rather than on City (from 0.019 to −0.010).

Cross-environment generalization matrix: R^2 obtained when the model trained on the row environment is evaluated, without refitting, on 300 fresh samples from the column environment. Diagonal entries (same-environment testing) are not expected to match the 20-run averages of Table 6 exactly because a different sampling protocol is used (Section 7.5); all entries are negative except City-trained/City-tested and Village-trained/City-tested (both R^2 = +0.019), and degradation is most severe for the Farm-trained model applied to City (R^2 = −60.7), showing that a model trained on one deployment density fails to transfer when applied to a different one.

Figure 5 presents the results of Table 7 as a heat map. Among 9 train/test datasets, only 2 were non-negatives: the City diagonal (City-trained, City-tested, R^2 = 0.019) and one off-diagonal (Village-trained City-tested, R^2 = 0.019). The Farm-trained model, on the other hand, is drastically miscalibrated (R^2 = -60.7 and -4.8, respectively), when applied to City or Village data, because the level of interference in the Farm, with sparse, long-range links, differs significantly from that in the City. In this way, it completely contradicts any proposal that a single lightweight predictor will be able to generalize across various deployment environments through the lens of this simulation model and feature set; environment-specific (re)training is indispensable, aligning with the concerns of domain-shift in the generalizability of models in the wireless literature [10].

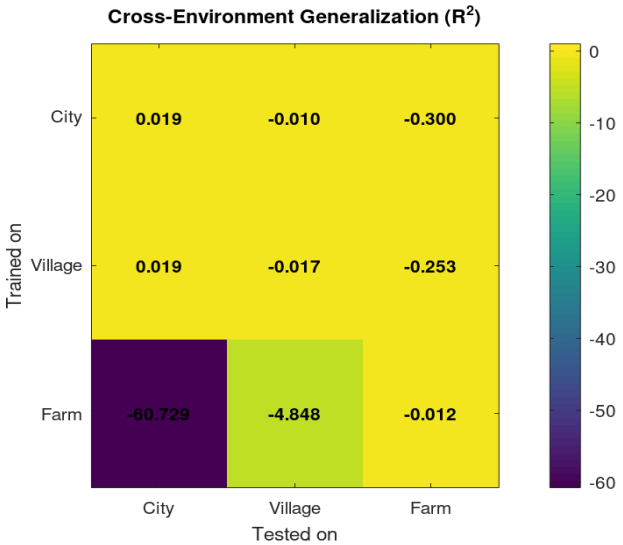


Figure 5. Cross-environment generalization.

7.6. Sensitivity to node density and channel count

This subsection tests whether the small energy improvement reported in Section 7.2 is specific to the baseline node density and channel count listed in Table 2, or whether it holds more generally. Using the same 20-run Monte Carlo protocol as Section 7.1, we sweep interferer node density from 0.5× to 3× the baseline value and, separately, the number of available sensing channels (1, 2, 4, 8), holding all other parameters fixed, and report the resulting percentage energy improvement of ProposedFull over StaticBaseline. The results of the two sweeps are shown in Figure 6 and Figure 7, respectively.

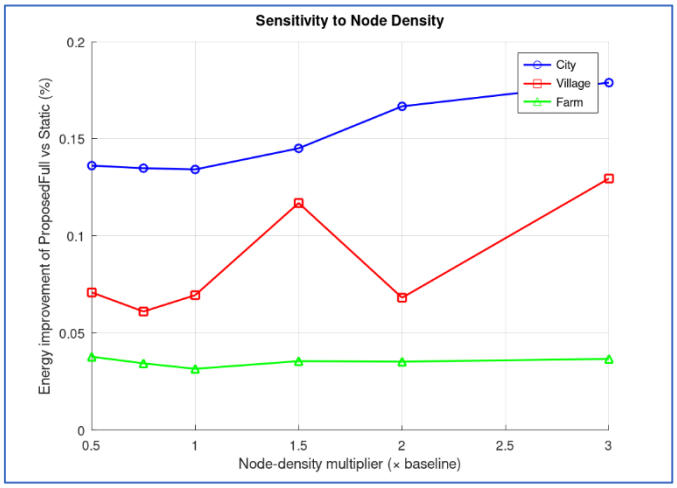


Figure 6. Sensitivity to node density.

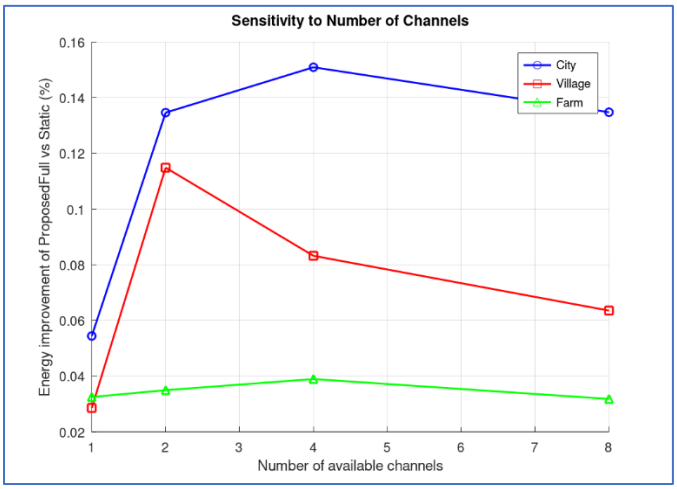


Figure 7. Sensitivity to number of channels.

Energy improvement (%) of ProposedFull over StaticBaseline as node density is scaled from 0.5× to 3× the baseline value in Table 2. The improvement stays within a narrow 0.03–0.18% band across the full density range in every environment, i.e., the small benefit reported in Section 7.2 is not an artifact of the specific baseline density but remains consistently small across a six-fold density range.

As shown in Figure 6, the City improvement ranges from 0.134% (0.5× density) to 0.179% (3× density); Village ranges from 0.061% to 0.130%; Farm stays within 0.031–0.038% throughout. The improvement increases weakly with density, as expected, because more interferers give power/channel adaptation more room to help, but the increase is marginal relative to the idle-energy floor established in Section 7.3. Figure 7 shows that the dependence on the number of channels is non-monotonic: City peaks at four channels, Village peaks at two channels, and Farm is essentially flat, so adding channels does not necessarily increase the energy saving in this model.

Energy improvement (%) of ProposedFull over StaticBaseline as the number of available channels is varied (1, 2, 4, 8). The relationship is non-monotonic: City peaks at 4 channels (0.151%) and declines at 8 (0.135%); Village peaks at 2 channels (0.115%); Farm is roughly flat (0.032–0.039%) across all channel counts. Thus, more channels do not necessarily translate into more energy saving in this model.

7.7. Runtime benchmark

Table 8 shows a runtime benchmark that compares the time spent on ML components (training and prediction) to the per-decision execution time for the decision-making strategies (Static, Reactive, PowerOnly, and Full) across the three scenarios. ML components are not time-consuming. Training is about 0.31–0.36 ms. The time taken to make one prediction is only 0.0200–0.0289 ms, which is fairly consistent with different environments. This implies that the ML parts contribute only slightly to the overall compute cost of a decision. Among the four approaches, Static and Reactive are clearly top performers in terms of per-decision speed, since they rely on simple fixed or only barely adjusted logic, respectively.

On the other hand, PowerOnly and Full are about an order of magnitude slower (2.5–3.4 ms per decision) as a result of the constrained search in Algorithm 1 performed on the set of discrete power levels (and for the Full scheme, the channels as well) at every decision.

Table 8. Runtime benchmark: ML overhead and per-decision execution time.

Environment	ML train (ms)	ML predict (ms/call)	Static (ms/decision)	Reactive (ms/decision)	PowerOnly (ms/decision)	Full (ms/decision)
City	0.34	0.0200	0.133	0.146	2.725	2.940
Village	0.31	0.0215	0.146	0.172	2.676	3.365
Farm	0.36	0.0289	0.200	0.199	2.841	2.555

On Farm, Full (2.555 ms) is faster than PowerOnly (2.841 ms), but on the other hand, in City and Village the ordering is the opposite: The city records Full at 2.940 ms compared to PowerOnly at 2.725 ms, and the village Full is measured at 3.365 ms versus 2.676 ms. This environment-dependent ordering suggests that the added cost of Full's channel-sensing step over PowerOnly is not fixed, but varies with how often the constrained search evaluates additional channel candidates, which in turn depends on the interference conditions of each environment. In summary, although the Full scheme gives the best energy and reliability results among the tested schemes (Tables 3–5), it incurs a small but finite computational overhead of a few milliseconds per decision, which is negligible relative to the 3600 s decision interval, while the ML overhead is negligible in all environments.

ML training and inference are sub-millisecond; the predictive schemes' constrained search over power levels and channels takes on the order of 2.5–3.4 ms per hourly decision, roughly 20× slower than the static/reactive schemes but still far below the 3600 s decision period, which supports, by direct measurement, the claim that the predictor/optimizer is lightweight relative to the decision interval. However, these values are not claimed to be representative of constrained embedded hardware.

The per-decision times of all schemes are compared on a logarithmic scale in Figure 8.

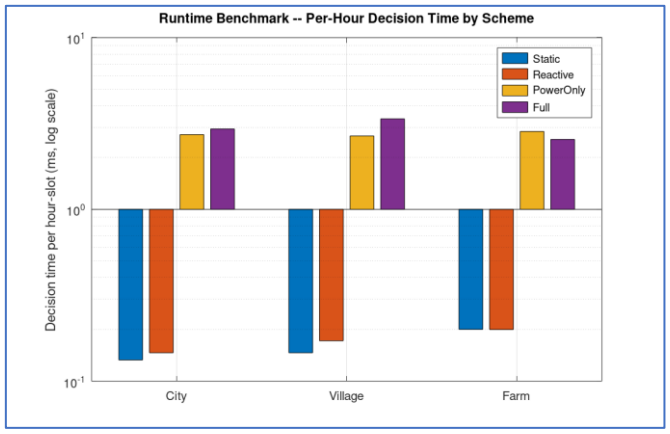


Figure 8. Runtime benchmark: per-hour decision time by scheme.

Per-hour decision latency by scheme (log scale), measured on the machine used for this study (indicative only, not a hardware-representative embedded benchmark).

8. Discussion and Limitations

The central finding of this study is a distinction between statistical significance and practical magnitude. The proposed predictive power-and-channel scheme reduces mean daily energy relative to a static baseline with $p < 0.001$ in every environment, and it is the only scheme that reaches exactly PDR = 1.000 with zero cross-run variance. However, the absolute reduction is 0.03–0.14% of total daily energy, because idle listening, constant across all four schemes by construction, accounts for 99.2–99.9% of the energy budget (Section 7.3). The substantial relative saving (17–22%) applies only to the transmit-energy component and does not carry through to total energy; energy-saving claims for power/channel adaptation should therefore always be stated relative to the total energy budget.

Three further limitations follow directly from the results in Section 7. First, the interference regressor's $R^2 \approx 0$ (Section 7.4) shows that temperature, humidity, and nominal transmit power are not sufficient features to predict interference under the Poisson-interferer-field model used here; the dominant source of variance is the random number and placement of interferers, which is not observable from these three variables. A node-density estimate, a recent-history feature (e.g., a moving average of past SINR), or a spatial/topology feature would likely

be necessary before claiming useful interference prediction. Second, the high-interference classifier's near-zero precision/recall means it should not be relied upon operationally in its current form; the constrained-optimization search in Eq. (13), which uses the point regression prediction directly rather than the classifier's binary flag, performs the effective adaptation in the proposed scheme. Third, cross-environment generalization is poor (Section 7.5): a model trained in one deployment density should not be deployed unmodified in another, which limits the "train once, deploy everywhere" appeal of a lightweight ML predictor and is consistent with known domain-shift effects in wireless ML more broadly [10].

The clearest actionable implication of the idle-energy dominance found here is that any future extension of this scheme aiming for energy savings beyond a few tenths of a percent needs to target idle-listening energy directly, for example through duty-cycling or a wake-up-radio mechanism [3], rather than, or in addition to, transmit-power and channel adaptation. Such a mechanism was not implemented or evaluated in this study and is identified as the most promising direction for future work.

Additional limitations: (i) the assumed IEEE 802.15.4 PHY/MAC parameters [11] are a modeling choice, not a validated hardware trace, and should be re-confirmed against whatever standard a hardware follow-up targets; (ii) the linear regression and logistic classifier are deliberately simple baselines and were not compared against more expressive models (e.g., gradient boosting or a small neural network), which is a natural next step now that the pipeline is reproducible; (iii) the runtime benchmark (Section 7.7) was measured on a general-purpose machine and is indicative only, not a substitute for measurement on constrained embedded hardware; and (iv) all results are simulation-based, and no field validation against a real IoT deployment has been performed, as stated explicitly in the abstract and methodology.

9. Conclusion

This paper presented a simulation-based, statistically validated evaluation of an ML-assisted power-and-channel adaptation scheme for IoT interference and energy management across three synthetic deployment densities. Using a physically grounded channel/energy model and a fully reproducible ML and optimization pipeline, we found that the proposed scheme achieves perfect packet delivery and a statistically significant but practically small (0.03–0.14%) reduction in total daily energy, because idle listening dominates the energy budget and is unaffected by power or channel adaptation. The results also show that the interference regressor has near-zero explanatory power from the assumed feature set, the high-interference classifier does not reliably detect positive cases, and cross-environment generalization is largely unsuccessful. These results actually limit rather than exaggerate the extent of a lightweight ML-assisted adaptation layer performance in this simulation model and suggest the idle-energy reduction and introduction of more predictive features as the immediate directions for future work. All simulation code, parameters, and random seeds used to produce every number in Section 7 are available for independent reproduction (see Data Availability Statement).

Author Contributions

Zeyid T. Ibraheem: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft. Nassma M. Ammar: Validation, Data curation, Visualization, Writing – review & editing. Mohammed Aboud Kadhim: Methodology, Supervision, Writing – review & editing. All authors have read and approved the final manuscript.

Competing Interest Statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability Statement

This study is entirely simulation-based; no field-collected or third-party dataset was used. All data reported in this article were generated synthetically by the simulation model documented in Sections 3–6, using the parameter values in Table 2 and the deterministic seeding rule described in Section 4.3. The complete simulation

code, configuration files, and fixed random seeds required to regenerate every number, table, and figure in Section 7 are available from the corresponding author, Zeyid T. Ibraheem (E-mail: zeyid.tariq@uoitc.edu.iq), and will be provided free of charge and without restriction within five working days of a request.

Statement on the Ethical Use of AI Tools

During the preparation of this revision, the authors used Grammarly to assist with language editing, proofreading, and checking the consistency of notation, abbreviations, and reference formatting. The tool was not used to generate data, run the simulations, perform the analyses, or produce the figures and results. The authors reviewed and edited all AI-assisted text and take full responsibility for the content of this manuscript.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgments

The authors thank the University of Information Technology and Communications and Middle Technical University, Baghdad, Iraq, for the institutional support provided during this work.

References

[1] Y. Hmamouche, M. Benjillali, S. Saoudi, H. Yanikomeroglu, and M. Di Renzo, "New trends in stochastic geometry for wireless networks: A tutorial and survey," *Proc. IEEE*, vol. 109, no. 7, pp. 1200–1252, Jul. 2021, doi: 10.1109/JPROC.2021.3061778.

[2] D. Garlisi, A. Pagano, F. Giuliano, D. Croce, and I. Tinnirello, "Interference analysis of LoRaWAN and Sigfox in large-scale urban IoT networks," *IEEE Access*, vol. 13, pp. 44836–44848, 2025, doi: 10.1109/ACCESS.2025.3550014.

[3] M. Q. Alazzawi, J. C. Sánchez-Aarnoutse, A. S. Martínez-Sala, and M. D. Cano, "Green IoT: Energy efficiency, renewable integration, and security implications," *IET Netw.*, vol. 14, no. 1, Art. no. e70003, 2025, doi: 10.1049/ntw2.70003.

[4] P. Gautam, M. Vakilifard, C. Bockelmann, and A. Dekorsy, "Cooperative interference estimation using LSTM-based federated learning for in-X subnetworks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Kuala Lumpur, Malaysia, 2023, pp. 1338–1344, doi: 10.1109/GLOBECOM54140.2023.10437515.

[5] P. Gautam, C. Bockelmann, and A. Dekorsy, "Interference prediction in unconnected in-X mobile 6G subnetworks using a data-driven approach," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Denver, CO, USA, 2024, pp. 2046–2052, doi: 10.1109/ICCWorkshops59551.2024.10615704.

[6] N. H. Mahmood, O. A. López, H. Alves, and M. Latva-Aho, "A predictive interference management algorithm for URLLC in beyond 5G networks," *IEEE Commun. Lett.*, vol. 25, no. 3, pp. 995–999, Mar. 2021, doi: 10.1109/LCOMM.2020.3035111.

[7] W. Bai, G. Zheng, W. Xia, Y. Mu, and Y. Xue, "Multi-agent deep reinforcement learning-based joint channel selection and power control method," *Comput. Electr. Eng.*, vol. 123, Art. no. 110147, 2025, doi: 10.1016/j.compeleceng.2025.110147.

[8] W. Bai, G. Zheng, W. Xia, Y. Mu, and Y. Xue, "Multi-user opportunistic spectrum access for cognitive radio networks based on multi-head self-attention and multi-agent deep reinforcement learning," *Sensors*, vol. 25, no. 7, Art. no. 2025, 2025, doi: 10.3390/s25072025.

[9] A. Pagano *et al.*, "A survey on massive IoT for water distribution systems: Challenges, simulation tools, and guidelines for large-scale deployment," *Ad Hoc Netw.*, vol. 168, Art. no. 103714, 2025, doi: 10.1016/j.adhoc.2024.103714.

[10] M. Akrout, A. Feriani, F. Bellili, A. Mezghani, and E. Hossain, "Domain generalization in machine learning models for wireless communications: Concepts, state-of-the-art, and open issues," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 3014–3037, 4th Quart. 2023, doi: 10.1109/COMST.2023.3326399.

[11] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015), 2020, doi: 10.1109/IEEESTD.2020.9144691.