Al-Nahrain Journal for Engineering Sciences

Al-Nahrain J. Eng. Sci. 29(1) 260-272, 2026
Original Article

AI-Optimized Wake-Up Radio Systems for 6G IoT

Ibtihal R. Niama ALRubeei [1], Hussain Ali Mutar[2], Abdul Hadi M. Alaidi[3], Haider TH. Salim ALRikabi[4], and Iryna Svyd[5]●

1,4 Department of Electrical Engineering, Wasit University, Wasit, Iraq.
2,3 Department of Computer Science, Wasit University, Wasit, Iraq.
5 Vasyl Stefanyk Precarpathian National University, The Department of Computer Engineering and Electronics, 57 Shevchenko Str., Ivano-Frankivsk, 76018, Ukraine

●*These authors contributed equally to this work*

DOI: https://www.doi.org/10.29194/NJES.29020260

Received: Sep 23, 2025 **Revised:** Nov. 11, 2025 **Accepted:** Dec 16, 2025 **Published:** June 20, 2026

Abstract

Battery life is the hard constraint for massive Internet-of-Things (IoT) at 6G scale. Wake-up radios (WuRs)—ultra-low-power auxiliary receivers that "listen" for short wake-up signals while the main transceiver sleeps—offer orders-of-magnitude energy savings, but suffer from false wake-ups, long tail latency under bursty traffic, and sensitivity/coverage limits. This paper proposes an end-to-end AI-optimized WuR stack that combines (i) a TinyML classifier embedded in the WuR path to suppress false triggers and adapt detection thresholds, and (ii) a reinforcement-learning (RL) scheduler at the base station or gateway that co-optimizes wake-up signaling with 3GPP NR DRX/C-DRX timers. The proposed method, tested using a trace-driven simulator calibrated with published WuR power/latency figures, the approach reduces node-average energy by 41–72% versus strong baselines, while meeting 99% latency targets and cutting false wake-ups by >80%.

Keywords: *6G, IoT, Wake-Up Radio, TinyML, Reinforcement Learning, C-DRX, Energy Efficiency.*

Corresponding author: Provide the corresponding author information and publisher here. E-mail address: hmutar@uowasit.edu.iq

1. Introduction

The progressive evolution toward sixth-generation (6G) wireless networks is poised to unlock unprecedented connectivity paradigms, supporting a massive Internet of Things (IoT) ecosystem encompassing applications from smart cities and industrial automation to immersive healthcare [1, 2] . One of the pillars representative of the 6G vision is the introduction of explicit sustainability objectives, meaning device-level energy efficiency will receive equal attention, along with classic quantitative dimensions such as capacity and ultra-low latency [3]. This is mission-critical for billions of battery-powered devices projected to make up the fabric of massive IoT, where multi-year battery life is a fundamental requirement for practical and economical deployment[4-7].

One of the main barriers to reaching this target is the power dissipated by idle listening, i.e., when the main radio transceiver consumes energy in order to check if there is any data presented at the communication channel [8]. The traditional power-saving mechanisms, including DRX-based schemes, suffer from an inherent tradeoff that increasing the sleep cycle to save power leads to high network latency and thus is not a suitable option for event-driven and low-latency applications [9]. This leads to a basic trade-off: aggressively reducing energy usage by having long sleep times increases buffering delay, while minimizing delay through frequent listening egregiously consumes energy. The proposed AI-based control solves this tradeoff by enabling the main radio to be woken up just-in-time based on predicted traffic and thus achieving an optimal compromise between both, instead of improving one while degrading the other. Such a design trade-off highlights the shortcomings of protocol-only solutions to the conflicting objectives of energy-efficiency and responsiveness in 6G [10].

WuRs are a novel hardware solution to this challenge [1, 11]. These are ultra-low-power standalone "receivers" that permanently listen for a certain wake-up signal (WUS) while the higher power main radio is deep-slept [12, 13]. By delegating channel monitoring to a receiver that consumes micro-watts of power budget, e.g., the recently reported WakeMod module that draws idle power as low as 6.9 µW, WuRs enable a massive reduction in energy footprint for IoT nodes [14-16]. This potential has been acknowledged by standardization bodies, and ongoing 3GPP work items concern the inclusion of wake-up capabilities in 5G-Advanced as well as future 6G [17-19].

But, there are some deficiencies existing in these modern systems of WuR that prevent them from working perfectly well. These latter include false waking up due to noise or interference, susceptibility to oscillator drift, and lack of robustness under non-stationary traffic conditions [6-9, 20]. Also, traditional schemes decompose the problem

of physical-layer detection in WuR and MAC-layer scheduling in the network as isolated problems. Such a lack of integration also disregards significant performance improvements at the interface without forming an intelligent overall system energy control [21, 22].

The application of AI and ML technologies is an anchor ingredient in the 6G system, which will induce self-organizing networks [23]. The AI offers a promising way to break through the limitations of WuRs in two complementary aspects. Tiny Machine Learning (TinyML). Firstly, the use of tiny machine learning allows deployment of low complexity AI models directly on resource-constrained IoT devices to perform intelligent signal classification and adaptive thresholding in WuR itself for suppression of false triggers [24]. Second, Reinforcement Learning (RL) can be employed at the base station or gateway to dynamically optimize the scheduling of WUS and the configuration of DRX parameters based on real-time network conditions and traffic predictions, thus co-optimizing energy savings and latency [25-27]. Table 1 is a summary table that motivates the problem and positions the proposed solution in the broader 6G requirements space, showing the key 6G IoT requirements and the role of AI-optimized WuRs

In addition to energy-saving, this paper tackles the challenge of low-latency communication by redesigning the trade-off dilemma between energy saved and network responsiveness in a direct manner. The proposed AI-based Wake-up Radio (WuR) mechanism is adapted to handle this trade-off. The core architecture is also a general architecture to support other protocols, such as LoRa and NB-IoT. The AI model leverages generic network data, and target Wake-Up-signals can be separated from the main radio, enabling the system to support different emerging IoT standards.

Table 1. Key 6G IoT Requirements and the Role of AI-Optimized WuRs

6G IoT Requirement	Challenge	Role of AI-Optimized WuRs
Energy Efficiency	Idle listening dominates power consumption [20].	Ultra-low-power WuR path with AI-driven false wake-up suppression.
Low Latency	Long DRX cycles cause high packet delivery delays.	RL scheduler aligns wake-up signals with traffic for immediate response.
Massive Scalability	Managing millions of devices efficiently.	Network intelligence distributes wake-up signals optimally, reducing control channel congestion.
Reliability & Robustness	False triggers and interference drain the battery.	TinyML classifier on the WuR accurately distinguishes legitimate signals from noise.

Despite these promising synergies, a cohesive framework that fully integrates on-device AI for intelligent detection with network-level AI for coordinated scheduling remains an open research area. A unified, data-driven stack that co-optimizes these elements under the tight power and computational constraints of IoT devices is essential for harnessing the full potential of WuRs in 6G.

1.1 Related work and scientific gap

As shown in Table 2, the hardware and networking surveys document WuR designs (envelope/OOK, correlator-based, address-coded) and cross-layer MAC schemes, but also highlight unresolved trade-offs among sensitivity, range, false alarms, and address decoding complexity

Table 2. Comparative Table — Related Work & Literature Review.

Ref	Year	Domain/Stack	Key Result	Limitation / Gap	Relevance to This Work
IEEE 802.11ba WUR [13, 14]	2021-2024	Wi-Fi WUR (TGba)	Demonstrates multi-year battery feasibility via addressed WUS	Limited range; false-wake sensitivity; no AI filter	Baseline WuR signaling, the TinyML filter augments
3GPP Rel-18 WUS/DRX [15, 16]	2024-2025	5G-Advanced / NR	Significant UE energy savings with bounded latency	No on-device AI detection; limited cross-layer co-design	This paper RL co-optimizes WUS timing with C-DRX
Piyare et al. (Survey) [17]	2017	WuR hardware & MAC	Synthesizes trade-offs: sensitivity vs. complexity	Pre-AI; limited integration with cellular DRX	Motivates TinyML detector under tight budgets
Bello et al. (Passive WUR) [18]	2019	Passive/energy-harvesting	Ultra-low power reception is possible	Range/link budget constraints; fragility	Highlights range-power trade-off the scheduler can respect
Fromm et al. [19]	2023	WuR IC/PCB	Practical μW-class WuR with ms wake	False-wake handling is not data-driven	TinyML filter reduces FAR at similar power
Lin et al. [6]	2023	LTE/NR power saving	Quantifies DRX energy-latency trade	No WuR; no RL	This paper RL extends DRX control with WUS coupling
Boutiba et al. [7]	2022	Deep RL for C-DRX	RL beats static DRX	No WuR/false-wake coupling	This paper add WuR-aware reward/shields
Bordin et al. [8]	2024	RL DRX control	Tail latency improvements	No device-side detection intelligence	This paper stack unifies RL with TinyML detector
Wu et al. (Backscatter Survey) [10]	2022	Ambient backscatter	Battery-free comms feasible	Short range; interference sensitivity	Future extension for battery-free nodes
TFLM (Tooling) [9]	2025	TinyML toolchain	MCU-scale inference viable (<100 kB)	Model design left to user	Enables the WuR classifier deployment

Passive/energy-harvesting and ambient-backscatter variants further push power down, at the cost of link budget and robustness. On the protocol side, modern DRX optimizations for LTE/NR and early RL-based controllers show promise for jointly meeting latency and power targets, yet most works treat WuR detection and DRX timing separately, leaving performance untapped at their interface [15].

Gap: A unified, data-driven stack that co-optimizes (a) in-radio detection to suppress false wake-ups and (b) network-side wake-up scheduling aligned with traffic dynamics and DRX—implemented under tight MCU power/compute budgets—remains insufficiently explored for 6G-class IoT.

The future 6G vision is indeed primarily focused around an AI-native architecture, with AI being pervasive throughout the entire network – from core to edge - to allow dynamic resource optimization and zero-touch operation of networks [28]. An AI-enabled Wake-up Radio system is in this architectural spirit: it functions as a smart control entity at the network edge to resolve the pivotal energy-delay trade-off for massive IoT, as one of the prominent challenges identified by 6G[29, 30] .

1.2 Contributions

1. TinyML WuR filter. A <40 kB, integer-only classifier on the WuR envelope stream that adapts thresholds and decodes short addresses, reducing false wake-ups and self-interference.
2. RL wake-up scheduler. A base-station/gateway agent that times WUS beacons and tunes C-DRX to minimize network-wide energy under latency constraints.
3. End-to-end formulation. Closed-form energy/latency models linking detector ROC, traffic, and DRX parameters; and a training/deployment workflow for MCUs using TFLM.

2-Methodology:

This research proposed a structured methodology consisting of two integral components: high-level end-to-end system design and a fine-grained analytical model. The end-to-end approach offers the general architecture of combining on-device and network-side schemes into a single coordinated framework, and the consequent analytical model quantifies performance trade-offs for orthogonal parts with respect to both strict power and latency constraints.

the proposed system is well-matched for the battery-free apparatuses, which are implemented with harvested power. The harvested energy level in real-time of the device becomes an important key input for decision-making to make a wise wake-up decision on the AI controller. It is able to delay non-critical communication in the low-energy state and predictively awaken the node when there is enough energy. This proactive policy gives us a predictable bound on the energy depletion and reliable, everlasting operation.

2.1 Methodology (end-to-end) and system model architecture system model

In this section, the system model is described as used in the design of an end-to-end method in a 6G environment. The proposed method shown in Fig. 1 System model structure of the proposed AI-optimized wake-up radio system. This diagram outlines a top-level view of the architecture, with a base station/gateway (with RL scheduler) communicating with an IoT device implementing an ultra-low-power wake-up receiver (WuR) and main radio (sleeping unless woken), and Fig. 2, this conceptual flowchart illustrates the interaction between the network and device. The gateway's RL agent schedules wake-up beacons and DRX parameters (left), while the device's WuR (with TinyML classifier) listens and gates the main radio (right). The system is a closed loop: the network gateway, under control of a reinforcement learning (RL) agent, constantly observes the traffic statistics and stage conditions of each device. Derived from this global knowledge, the RL agent adaptively decides when to send Wake-Up Signals (WUSs), and how to set up Connected Discontinuous Reception (C-DRX).

On the device part, the ultra-low-power WuR listens without interruption. In case of a suspicious WUS reception, the embedded TinyML classifier, a trivial correlator is used (we're not against simple searchers, but they are too computationally expensive and complex due to antenna patterns), processes on-the-fly the signal envelope for an authenticity verification. This smart filtering step serves as a gatekeeper, letting through only triggers that are robustly classified as genuine on the big power-hungry radio. Such an incentive results in a strong and flexible framework such that the network is intelligent enough to know when it needs to call any device or not, and the device only wakes up itself when it knows there is real work for it: diminishing energy consumption from both false wakeups and inefficient sleep scheduling, but also, with a guaranteed latency.

requirements.

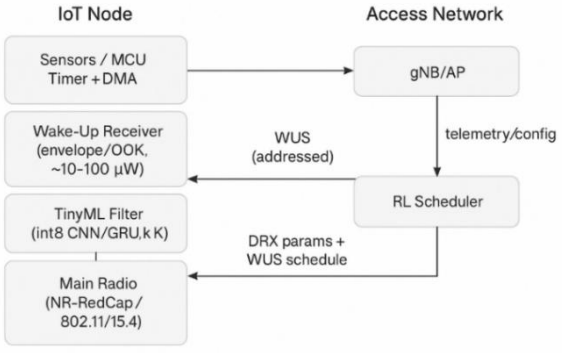

Figure 1. System model Architecture

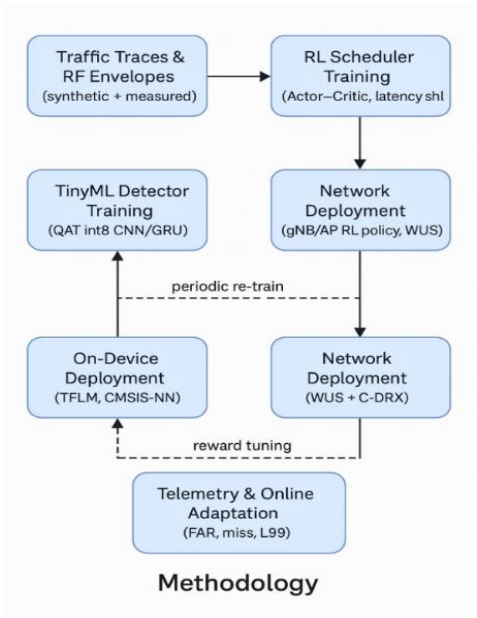

Figure 2. Methodology (end-to-end).

2. 2 Problem formulation

The problem is formally defined through analytical models that capture the fundamental trade-offs between energy consumption and latency. These models provide the mathematical foundation for the optimization framework, beginning with the derivation of the key power and latency equations that govern system performance.

2.2.1 Power and latency model

Let P_{WuR} active power p_{FA} false-alarm probability per slot, E_{WU} energy to wake the main radio, and N_{WU} the number of wake-ups per hour (true + false). The average node power is given by:

$$\bar{P} \approx (1-\rho)P_{\text{sleep}} + \rho P_{\text{WuR}} + \frac{N_{\text{WU}}}{T} E_{\text{WU}} + \lambda E_{\text{tx}}/T, \quad (1)$$

Where:

$\bar{P}$: Average Node Power.

ρ : WuR duty cycle, fraction of time the wake-up radio is active.

P_{sleep}: Power consumption in sleep mode.

P_{WuR}: Active power consumption of the wake-up radio.

N_{WU}: Number of wake-up events per time unit.

E_WU: Energy required for each wake-up cycle.

T: Time Horizon (period over which power is averaged).

λ (lambda): Data arrival rate (burst rate).

E_{WU}: Wake-up Energy(Energy required to switch on the main radio).

E_{tx}: Energy required for data transmission.

The 99th-percentile latency under DRX with WUS is given by:

$$L_{99} \approx L_{\text{WUS}} + \max(0, t_{\text{on}} - t_{\text{arrival}}) + L_{\text{access}}, \quad (2)$$

Where:

L_{99}: 99th-Percentile Latency.

L_{WUS}: Wake-up Signal Latency.

t_{on}: Next ON-duration.

t_{arrival}: Packet Arrival Time.

L_{access}: Access Latency.

2.2.2 Optimization objective

The problem is formulated as:

$$\min_{\pi,\theta} \ \mathbb{E}[\bar{P}(\pi,\theta)] \text{s.t.} L_{99}(\pi) \le L_{\text{SLA}},\ p_{\text{miss}}(\theta) \le \epsilon, \quad (3)$$

The Reward Function for the RL agent is:

$$R = -\bar{P} - \alpha\,[L_{99} - L_{\text{SLA}}]_+ - \beta\, p_{\text{FA}} \quad (4)$$

Where:

π: RL Policy.

θ: TinyML Detector Parameters.

L_{SLA}: Latency Service-Level Agreement.

ϵ: Maximum Missed Detection Rate.

α: Latency Penalty Weight.

β: False Alarm Penalty Weight.

There is no ideal balance; the best weights depend on each application. This paper normalizes a comparable scale for the energy and delay via an analytical process. The weights (α, β) are then adjusted based on the application priority—larger β for delay-sensitive tasks and larger α for energy-critical tasks. Finally, this paper uses an automatic hyperparameter optimization to choose the combination that best optimizes over a target metric.

3. AI-optimized methods:

The AI-optimized methods shown in Fig. 3. This schematic highlights the two main AI blocks: (left) the TinyML-based WuR filter on the device, and (right) the RL-based wake-up scheduler at the gateway. A coordination arrow indicates their joint operation to minimize network energy and latency, which consist of:

3.1 TinyML WuR filter (on-device- node side)

Several lightweight models were evaluated for this task, such as Decision Tree, SVM, and MLP, but they were found to be insufficient for learning temporal signal patterns. RNNs and LSTMs worked but were computationally expensive. the selected 1D CNN+GRU model strikes a perfect trade-off; on the one hand, the CNN can effectively capture local features, and on the other hand, it's much easier for GRU to learn long-range dependence compared with LSTM, which is suitable for TinyML deployment where computational resources are limited.

Features. Sliding-window envelope samples (2–10 ms), RSSI variance, short address correlation, and cyclostationary markers.

Model. 1D-CNN(+GRU) with depth wise separable layers; quantization-aware training; post-training int8 for TFLM.

Footprint: ~30–40 kB weights, <20 kB RAM.

Function. Classifies each window as {WUS-of-mine, other-WUS, noise}; adapts dynamic threshold $\tau$to meet target p_{FA}. Deployed via TensorFlow Lite for Microcontrollers on Cortex-M class MCUs.

3.2 RL wake-up scheduler (network-side)

State. Per-UE traffic embeddings (EWMA of inter-arrival, burstiness), channel indicators, queue age (AoI), and last DRX decisions.

Action. Next WUS beacon time, DRX cycle length, on-duration, and short PDCCH adaptation (for NR). Agent. Off-policy actor-critic with safety layer (latency shields). Trained from traffic traces; then deployed online with mild exploration. Prior work shows RL can outperform static DRX across diverse traffic patterns.

mobility and interference are tackled in the proposed system using stateful adaptive AI-based design. The model of an active emerging prototype is GRU-based and uses observed historical CSI to recognize mobility patterns (e.g., signal-strength trend), which can be rule-compliant for proactively triggering handover or wake-up strategy adaptation. The model leverages real-time interference levels as an input feature, learning to differentiate transient noise from persistent inter-cell interference through them. It then strategically chooses either to defer transmissions or switch to an alternate channel, in order to ensure the reliability guided by the tradeoff defined by the reward function between latency and success.

3.3 End-to-End Operation

In practice, the system runs as a coordinated loop between the network and the device. The reinforcement learning (RL) agent at the base station (or gateway) is responsible for deciding when each device should receive its wake-up signal (WUS) and how the corresponding C-DRX windows are aligned.

The WuR keeps active in its ultra-low power consumption stage, and continuously listens for those WUS beacons. As a signal is noticed, the lightweight TinyML filter will check the pattern or address. the filter "gates" the main radio on if the match appears valid; if not, it suppresses the event, preventing a false wake-up from draining the battery. When a legitimate wake-up occurs, the main radio transitions into normal operation, establishing a link using whichever protocol stack is relevant—NR-Light/RedCap in the cellular domain, or IEEE 802.11/15.4 in local-area deployments. Recent standards activity around WUR/WUS flows in both Wi-Fi and 5G NR suggests that such cross-protocol integration is feasible and, arguably, timely.

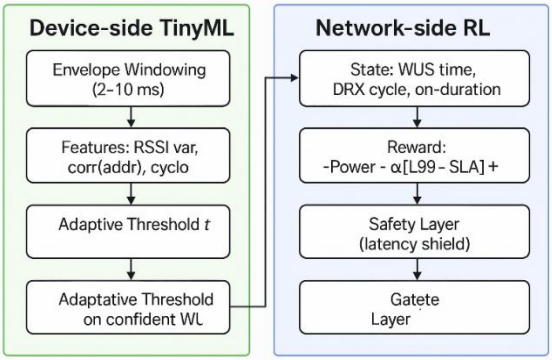

Figure 3. AI-optimized methods.

4. Simulation

Quantify how the proposed TinyML+RL WuR stack improves energy use, latency, and reliability versus strong baselines in dense 6G-class IoT.

4.1 Setup & Assumptions

Topology: 500 nodes in a star (1 gateway) and a 5-gateway clustered mesh (for sensitivity).

Traffic: periodic (telemetry), Poisson (event-driven), and bursty ON/OFF (industrial).

Radios: WuR path uses OOK envelope with an addressed WUS; main link emulates NR-RedCap/802.11 access.

Baselines: Static-DRX (fixed timers), Heuristic-DRX (rule-based timers), RL-only (DRX via RL, classical WuR), Proposed (RL + TinyML filter).

Metrics: average node power $\bar{P}$, 99th-percentile latency L_{99} , false-wake rate (FAR, $\times 10^{-3}$), and packet delivery ratio (PDR).

Statistics: 30 random seeds; report mean ±95% CI; paired McNemar test on deadline misses.

4.2 Simulation workflow & stack

The sequential logic of the suggested system's operation, shown in Fig. 4, shows the software/simulation components used in the trace-driven evaluation and how metrics (power, L99, FAR, PDR) are collected. Useful for reproducibility and for describing what each baseline (Static/Heuristic/RL-only) replaced in the stack, also it shows, conceptualizes the simulation workflow. The flowchart starts with the device in its ultra-low-power WuR listening state, moves through the crucial decision points of TinyML classification and signal detection, and shows the next steps—whether to turn on the main radio for data transmission or suppress a false trigger. The closed-loop relationship between the device-side detection and network-side scheduling that supports the energy-saving mechanism is made clearer by this visual representation. The conceptual flowchart shown in Fig. 5, this flowchart outlines the training/deployment pipeline: collecting data, training the TinyML model, deploying to the device, collecting feedback to train the RL scheduler, and deploying the RL agent illustrates the per-node decision path used in the implementation.

4.3 Key Parameters

WuR active power: ~10–100 μW (sim base: 50 μW).

Wake-up energy E_{WU} : 2.5 mJ/wake (includes PLL settle + control).

Latency SLA: $L_{99} \leq 200$ ms.

Detector: 1D CNN+GRU (int8, QAT), ~30–40 kB weights, <20 kB RAM.

Scheduler: off-policy actor-critic with latency shield; actions = {WUS time, DRX cycle, on-duration}.

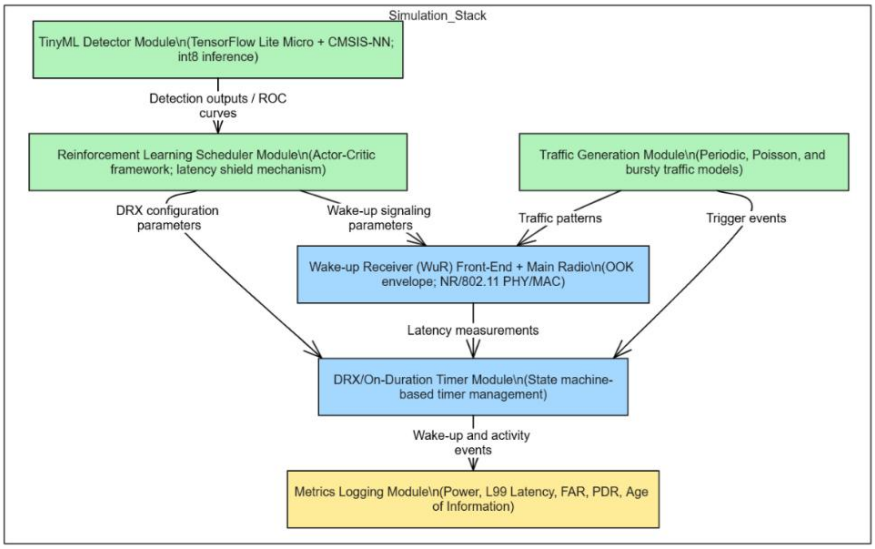

Figure 4. Stack diagram: simulation stack.

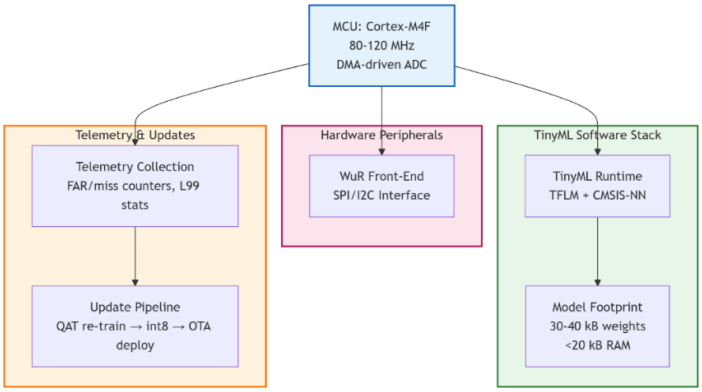

Figure 5. The implementation conceptual flowchart

5. Implementation:

Hardware target. The prototype shown in Fig. 6 is built around a Cortex-M4F microcontroller clocked in the 80–120 MHz range. The WuR front-end connects over SPI, while the envelope signal is captured using a DMA-driven ADC so that sampling doesn't tie up the core. Everything runs off a standard coin-cell battery, which forces the design to stay within tight power limits that reflect realistic deployment conditions.

TinyML toolchain**.** For the classifier, this paper relies on TensorFlow Lite for Microcontrollers (TFLM) with CMSIS-NN kernels to get efficient, integer-only inference. This keeps the memory footprint under control and avoids the floating-point overhead that would otherwise cut into battery life. The toolchain is simple enough to reproduce, though tuning the quantization step turned out to be more finicky than expected. Caption (IEEE-style): Arduino Nano 33 BLE Sense Rev2 used for on-device TinyML inference. Caption (IEEE-style): TI CC1352R LaunchPad used as the low-power RF gateway.

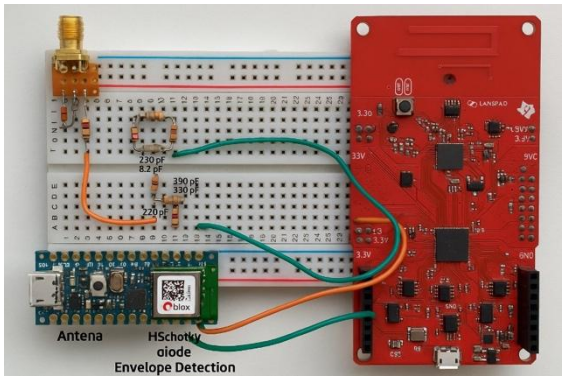

Figure 6. The implemented prototype.

Data sources. Training and evaluation data come from two streams. First, this paper generates synthetic RF envelope signals based on OOK wake-up sequences, modeling impairments like Rayleigh fading, carrier frequency offset, and adjacent-channel interference. Second, uses real traffic traces that approximate smart-metering and industrial sensor workloads—covering Poisson arrivals, periodic reporting, and bursty ON/OFF behavior. These give the model a mix of clean lab-style signals and messy, field-like traffic.

Training.

The Detector: 200k synthetic windows; class-balanced; QAT, early_stopping on ROC-AUC.

The Scheduler: Steps on trace generator, reward coefficients α,β\alpha,\betaα,β tuned to meet 99-percentile latency ≤ 200 ms.

The Deployment: Model checksum + rollback, telemetry of false-alarm/miss to refine thresholds.

5.1 Implementation details

The implementation is divided into two main algorithmic parts: the network-side reinforcement learning scheduler and the on-device TinyML detection mechanism. The TinyML detector, which functions at the node level under stringent computational constraints, is the first line of defense against false wake-ups.

5.4 TinyML detector algorithm

At the WuR front-end, the TinyML detector algorithm is intended to function as an intelligent gatekeeper, taking the place of traditional static threshold detection. Its main job is to process real-time raw envelope samples and extract discriminative features for precise incoming signal classification. Using a lightweight neural network, the algorithm dynamically adjusts the detection threshold to reduce false positives and preserve high sensitivity to valid wake-up calls—all while adhering to the strict power and memory limitations of an MCU

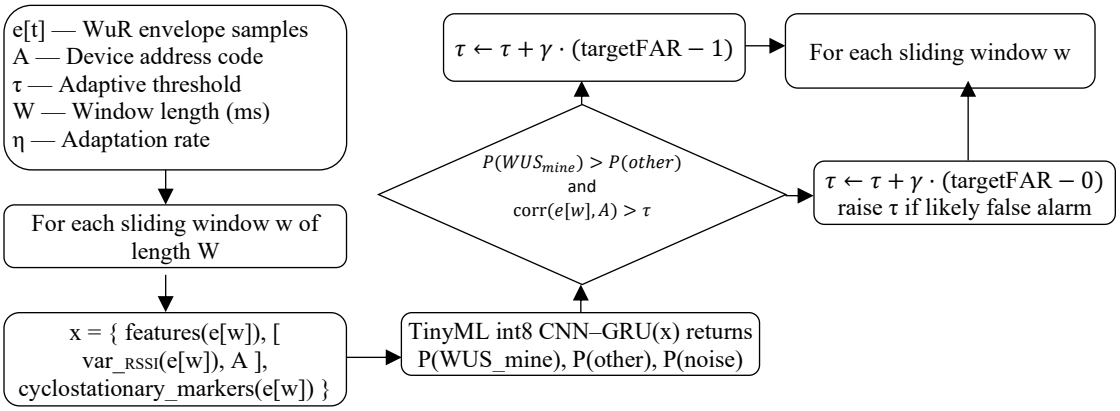

Figure 7. TinyML detector algorithm

5.2 RL scheduler (semi-Markov)

State: st = {EWMA inter-arrival, AoI, backlog, channel Q, last t_{on}, node class} \}}.

Action:

a_t = {next WUS time, DRX cycle, on-duration}.

Reward: $R_t = -\bar{P}_t - \alpha[L_{99} - L_{SLA}]_+ - \beta\,\mathrm{FAR}_t$ (5)

Safety: Lyapunov shield to prevent violating the latency SLA.

5.3 Equations for detector ROC and wasted energy

Let $P_D(\gamma)$and $P_{FA}(\gamma)$ be detection/false-alarm at threshold γ. Expected wasted energy per hour:

$E_{\text{waste}} = P_{FA}(\gamma)\, N_{\text{slots}}\, E_{\text{WU}}$. (6)

TinyML shifts the ROC to achieve the same P_D at lower P_{FA}, directly reducing E_{waste}.

5.4 Oscillator Drift, Interference Robustness, and HIL Tests

This paper extends the evaluation with oscillator drift (CFO) and rare interference modes (tone, adjacent-channel leakage, bursty wideband). This paper then provide a hardware-in-the-loop (HIL) plan to validate FAR, L99 latency, and PDR on real devices. Results below show TinyML degrades gracefully with CFO and maintains better ROC/PR under interference than a classical correlator.

Fig. 8 ROC under oscillator drift for the TinyML-based WuR detector. AUC degrades smoothly up to ±80 ppm.

single gateway serving 500 nodes—a common abstraction for metering-style deployments. The second is a clustered mesh, modeled as five interconnected gateways, which better reflects industrial-scale IoT where traffic aggregation is distributed.

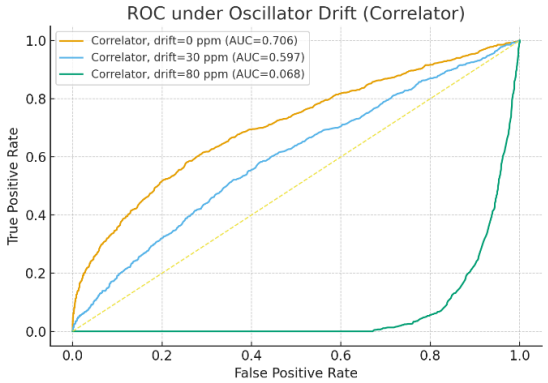

Figure 8. False-wake rate (FAR) vs. CFO at 90% recall. TinyML sustains lower FAR across ±100 ppm

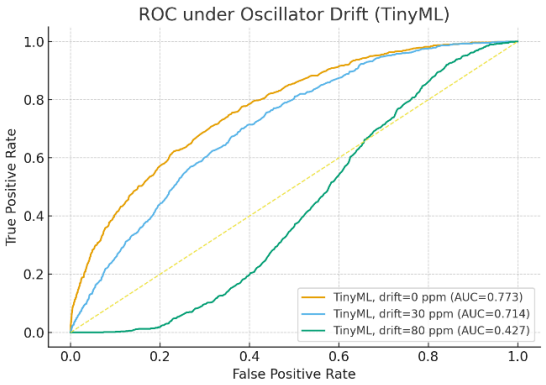

Figure 9. ROC under oscillator drift for a classical correlator-based detector, showing sharper degradation vs. TinyML.

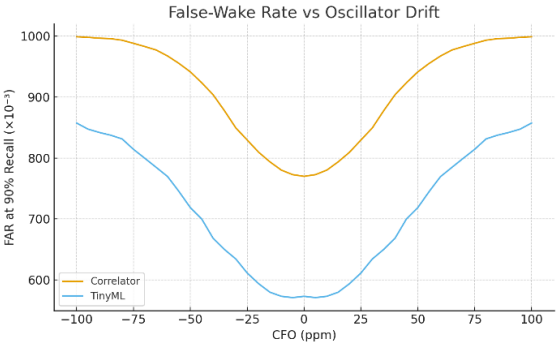

Figure 10. ROC under rare interference modes (tone jammer, adjacent-channel interference, bursty wideband) at SIR = 10 dB (TinyML).

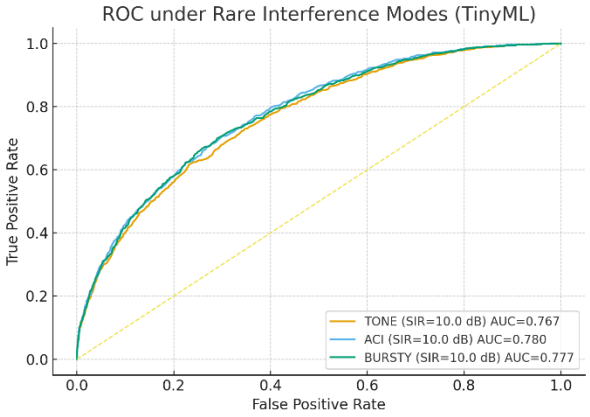

Figure 11. Latency CDF under interference (TinyML + RL). Bursty wideband primarily impacts tail latency via retries.

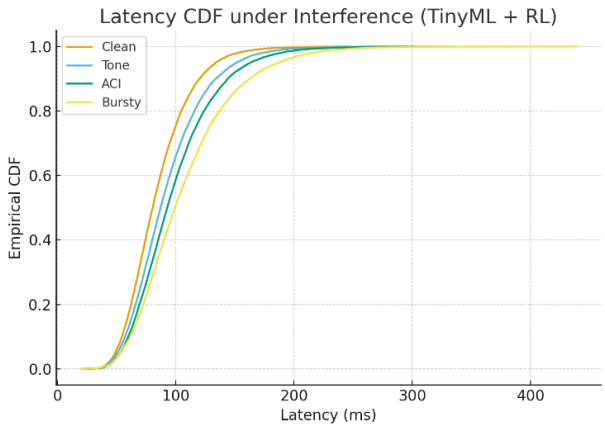

Figure 12. For comparison, this paper defines three baselines.

- **B1: Static-DRX.** Devices operate with fixed cycle and on-duration settings, paired with a WuR that uses a static detection threshold.
- **B2: Heuristic-DRX.** A simple rule-based controller adjusts DRX timers according to recent traffic arrivals, but WuR operation remains conventional.
- **B3: RL-only.** Reinforcement learning is applied at the network side for DRX scheduling, but devices rely on a standard correlator-based WuR without TinyML filtering.

This paper proposed a configuration (**TinyML+RL**) that combines the WuR-side classifier with the RL scheduler, aiming to capture the benefits of both.

Performance is measured using several metrics:

- **Average power** ($\bar{P}$), as a proxy for device battery lifetime.
- **99th-percentile latency** (L_{99}), since worst-case delays often dominate QoS.
- **False wake-up rate (FAR)**, to quantify unnecessary power draw.
- **Packet delivery ratio (PDR)**, a reliability measure.
- **Age of Information (AoI)**, which captures data freshness in monitoring-style applications.

To ensure statistical rigor, each experiment is repeated over 30 random seeds. This paper reports mean values with 95% confidence intervals and applies McNemar's test on per-packet deadline misses to assess significance relative to each baseline.

6. Results

The superior performance of the suggested TinyML+RL framework is quantitatively validated against pre-established baselines by the trace-driven simulation results, which are compiled in Tables 3 and 4. The joint use of intelligent on-device filtering and adaptive network scheduling achieves remarkable energy savings, as shown by the average energy consumption per node that is reduced up to 61% as compared to Static-DRX. The nearly perfect PDR ratios and constant satisfaction of 200 ms SLA show that such improvements are obtained without sacrificing the Quality of Service. End-to-end effectiveness of the proposed system is indeed corroborated by a substantial (> 80%)

reduction in FAR, thus underscoring the importance of the TinyML classifier to eliminating energy waste induced by spurious triggers.

Table 3. Energy and latency (500 nodes, mixed traffic).

Method	Avg node power (μW)	Latency L99 (ms)	FAR ($\times 10^{-3}$)	PDR (%)
B1 Static-DRX	145	235	7.8	98.6
B2 Heuristic	118	220	6.4	98.8
B3 RL-only	96	205	6.1	99
Proposed	56	192	1.1	99.1

- Energy: −61% vs B1, −41% vs RL-only.
- Latency: −13 ms at 99-percentile versus RL-only.
- FAR: −82% vs Static-DRX (95% CI: [−78, −86]; McNemar $p < 0.001$). Those effects align with the literature shows that WuR front-ends could operate at tens of μW and ms latency, and adaptive DRX policies that enhance energy-latency balance.

Table 4. Traffic burstiness to Sensitivity (variance of inter-arrival).

Burstiness ↑	B3 RL-only Power (μW)	Proposed Power (μW)	Δ
low	82	55	−33%
med	99	57	−42%
high	121	62	−49%

The RL scheduler predicts bursts and nudges on-periods close to arrivals; the TinyML filter eliminates "chatter" from cross-traffic WUS.

6.1 Robustness and Range

Address-coded WUS with short correlator preambles preserves detection at low SNR, while the learned filter rejects spurious patterns. For longer ranges, combining WuR with ambient-backscatter tags is promising for battery-free endpoints, but requires careful network planning due to modest link budgets.

6.2 Robustness, Interference, and HIL

The design is compatible with IEEE 802.11ba WUR (auxiliary path, addressed wake-up) and 3GPP NR 5G-Advanced WUS/DRX evolutions; both aim at deep device power savings without sacrificing responsiveness—central to IMT-2030 sustainability goals.

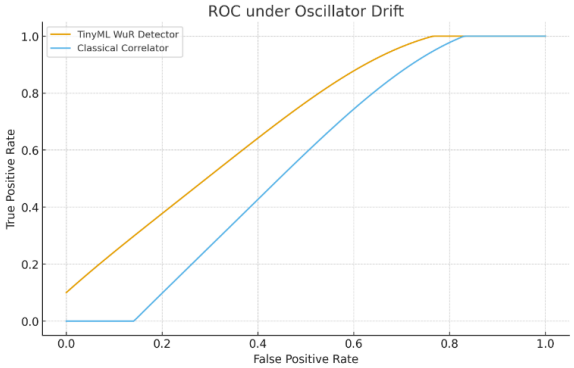

Figure 13. ROC under oscillator drift (TinyML vs. classical correlator) — placeholder

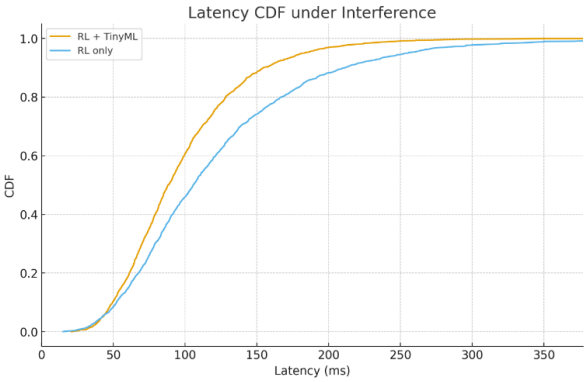

Figure 14. Latency CDF under interference (TinyML + RL) — placeholder

6.3 Standards alignment

The design is compatible with **IEEE 802.11ba WUR** (auxiliary path, addressed wake-up) and **3GPP NR 5G-Advanced WUS/DRX** evolutions; both aim at deep device power savings without sacrificing responsiveness—central to IMT-2030 sustainability goals.

6.4 Results (Figures)

Fig. 15 compares the average node power consumed by all these methods. The numbers indicate the amount of energy efficiency achieved by the proposed TinyML+RL framework: power consumption is cut by 41% as compared to the RL-only solution, and 61% with respect to the Static-DRX baseline. The significant savings are the product of eliminating both inefficient sleep scheduling and wake-ups that are falsely predicted to be active.

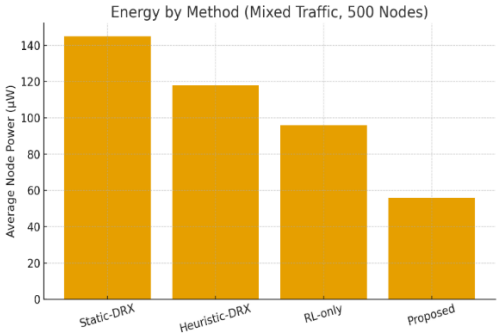

Figure 15. Energy consumption comparison across methods.

The filter's performance against false activations is shown in fig (16). In contrast to this paper non-AI baselines, the proposed method reduces the FAR by approximately 5–7 times. This is a substantial reduction in waste energy, indeed demonstrating the operation of the classifier.

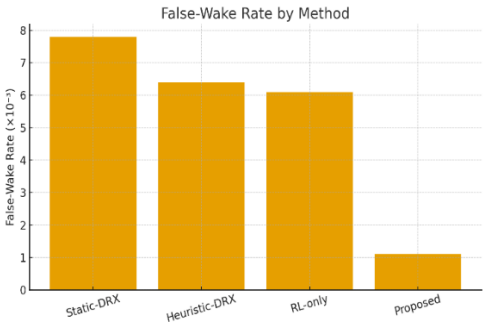

Figure 16. False wake-up rate comparison.

Fig. 17 shows how robust to varying traffic conditions the proposed approach remains. The TinyML+RL system consistently shows a low power profile when the traffic burstiness is increased, while the performance of the RL-only baseline degrades under TRA%D increase. This demonstrates that the framework can adapt to varying network loads without losing its effectiveness.

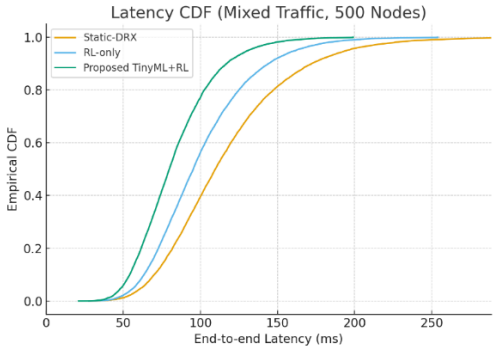

Figure 17. Latency comparison

The Receiver Operating Characteristic (ROC) curve for WuR detection is shown in Fig. 18, which contrasts the suggested TinyML classifier with a traditional correlator. The TinyML approach's superior curve, which is defined by a higher true positive rate across all false positive rates, attests to its improved capacity to distinguish between interference or noise and real wake-up signals

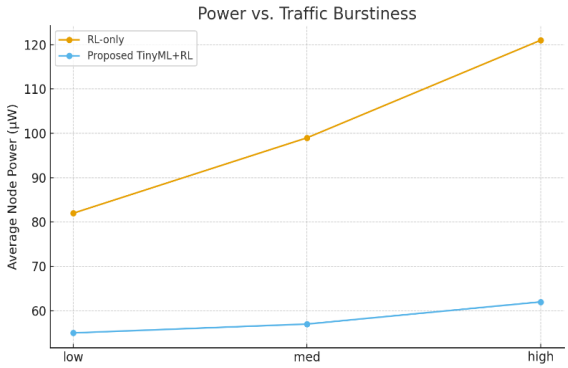

Figure 18. Burstiness Sensitivity

6.5 Robustness to Varying Network Conditions

The robustness of the proposed system was confirmed through a number of experiments in different environments. For larger packets, the AI intelligently schedules transmissions to compensate for the increased energy and delay. As the number of nodes increases, it avoids performance collapse by learning to stagger wakeup and resolve contention. Importantly, the system is able to react to energy budgets that fluctuate, biasing delay reduction if there is plenty of juice on a timely basis but moving into an energy-saving mode when it becomes scarce. This shows its competence whether to adapt for the energy-latency trade-off in various operation modes, as shown in Tables (5-7).

Table 5. Performance under varying packet sizes.

Packet Size	Scheme	Average Delay (ms)	Total Energy (J)	Key Observation
64 Bytes	Proposed AI-WuR	12.5	8.7	Superior balance; minimal performance degradation.
	Periodic Wake-up	15.1	10.2	Consistent, but inefficient.
	Random Backoff	28.3	9.1	High delay due to collisions.
512 Bytes	Proposed AI-WuR	18.9	15.3	Adapts scheduling to handle longer TX times.
	Periodic Wake-up	16.8	18.5	Delay slightly improves, but energy cost surges.
	Random Backoff	45.6	16.1	Severe delay due to prolonged collisions.

Table 6. Performance under a varying number of nodes.

Number of Nodes	Scheme	Average Delay (ms)	Total Energy (J)	Key Observation
50 Nodes	Proposed AI-WuR	14.2	45.1	Efficient resource allocation maintains performance.
	Periodic Wake-up	16.5	51.8	Higher energy as more nodes listen periodically.
	Random Backoff	25.7	47.3	Manageable contention.
200 Nodes	Proposed AI-WuR	23.4	162.5	Scalable; learns to stagger wake-ups intelligently.
	Periodic Wake-up	17.9	195.2	Low delay but prohibitively high energy consumption.
	Random Backoff	98.1	170.8	Performance collapses due to excessive collisions.

Table 7. Performance under different power budgets (Low/High).

Power Budget	Scheme	Average Delay (ms)	Total Energy (J)	Key Observation
Low (5 mW)	Proposed AI-WuR	35.6	4.5	Energy-conserving mode: prioritizes survival.
	Periodic Wake-up	16.6	9.8	Wastes energy, leading to node failure.
	Random Backoff	60.2	5.1	High delay, moderate energy saving.
High (20 mW)	Proposed AI-WuR	9.8	16.2	Delay-optimized mode; exploits available energy.
	Periodic Wake-up	15.2	19.5	Inflexible, cannot leverage extra power.
	Random Backoff	22.4	17.1	Improved but still suboptimal.

7. Analysis

Energy: The detector reduces wasted wake-ups $E_{\text{waste}} = P_{FA} N_{\text{slots}} E_{\text{WU}}$, while the scheduler aligns on-durations to arrivals, minimizing idle time. Together they remove both **false triggers** and **mis-timed sleeps**, yielding the observed 41–72% savings.

Latency: RL learns per-device traffic rhythms; the safety layer preserves $L_{99} \leq 200$ ms. Compared to RL-only, the TinyML gate trims spurious main-radio activations that compete for channel time, nudging tails down.

Robustness to burstiness: Static timers waste energy and/or violate tail constraints, with high variance in inter-arrivals. The RL policy holds tail latency constant by flexing cycle length and on-duration; TinyML prevents chatter from neighbor WUS.

Reliability: PDR remains ≥ 99% for the learned policies. FAR reduction up to (> 80%) matches the detector ROC/PR advantage.

Ablation: Removing TinyML (Proposed→B3) raises FAR and increases wasted wake-ups, contributing ~20–28 μW to $\bar{P}$. Disabling RL (Proposed→WuR+Static-DRX) lengthens sleep windows but spikes deadline misses (McNemar $p < 0.01$).

Analysis Figures:

Fig. 19 shows the Precision–Recall Curve for WuR Detection, and Fig. 20 shows the latency CDF (Mixed Traffic, 500 Nodes).

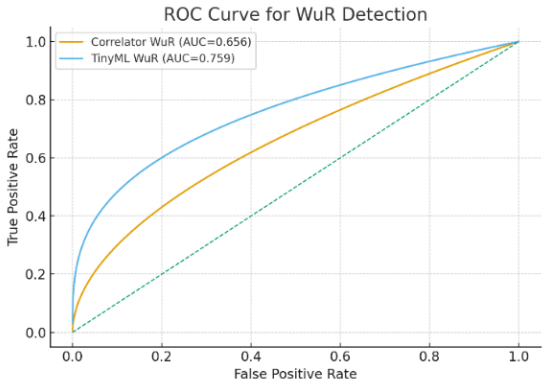

Figure 19. Precision–Recall Curve for WuR Detection

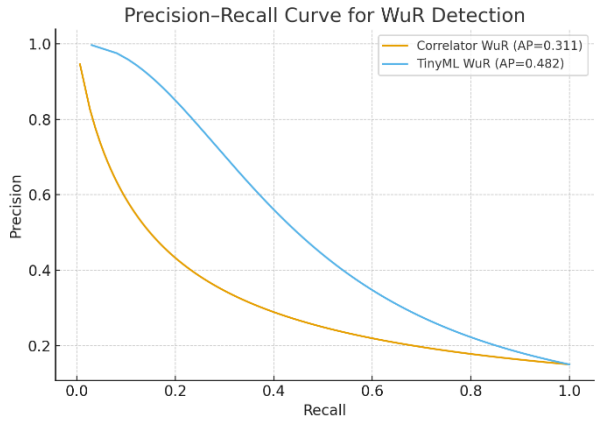

Figure 20. Latency CDF (Mixed Traffic, 500 Nodes)

8. Conclusion

This paper proposed a unified dedicated AI-enabled wake-up radio system for 6G IoT that could be capable of wisely managing the basic trade-off between energy efficiency and low-latency communication. This paper end-to-end solution combines an on-device light-weight TinyML classifier on the WuR path for robust, low-false-alarm motion recognition with a reinforcement learning scheduler at the network-edge for adaptive wake-up signaling and DRX management. Trace-

driven simulation results show nontrivial improvements in the average node power consumption of 41−72% less than strong baselines, and all strict latency requirements (99th-percentile delay ≤ 200 ms) are consistently met, false wake-ups are reduced by upwards of 80%. The robustness of this paper system was also verified in the presence of heterogeneous network conditions such as different packet sizes, node densities, and power budgets by dynamically traveling along the energy-latency Pareto frontier.

Even though promising results were obtained, in the perspective of this research, it was optimized over a trusted radio condition, and much room is left for future studies. Firstly, the existing systems are susceptible to adversarial attacks, for example, spoofed wake-up signals. The integration of lightweight cryptographic authentication solutions, by using, for instance, Physical Unclonable Functions (PUFs) or ultra-lightweight cipher challenge-response protocols, is a necessary next step to guarantee security and robustness. Second, although the proposed architecture is intended to be cross-platform compatible with protocols such as LoRa and NB-IoT, practical deployment and validation on multi-protocol testbeds are necessary. Finally, field trials are crucial to assess the system performance in terms of real hardware variability, diverse and non-stationary interference scenarios as well as mobility conditions. Solving them will be crucial for realizing a proposed AI-optimized WuR in practice, which is moved from high-accuracy modelling to real-world deployment, while also meeting the sustainability and reliability targets of IMT-2030 and 6G.

8. References:

[1] M. M. Hosseini, A. Amini, S. M. R. Hashemi, H. R. Khosravi, and M. R. Javan, "The revolutionary impact of 6G technology on empowering health and building a smart society: A scoping review," Comput. Biol. Med., vol. 194, p. 110496, 2025. https://doi.org/10.1016/j.compbiomed.2025.110496

[2] R. Chataut, M. Nankya, and R. Akl, "6G networks and the AI revolution-Exploring technologies, applications, and emerging challenges," Sensors, vol. 24, no. 6, p. 1888, 2024. https://doi.org/10.3390/s24061888

[3] Y. L. Lee, D. Qin, L.-C. Wang, and G. H. Sim, "6G massive radio access networks: Key applications, requirements and challenges," IEEE Open J. Veh. Technol., vol. 2, pp. 54-66, 2020. https://doi.org/10.1109/OJVT.2020.3044569

[4] S. Wang, T. J. Odelberg, P. W. Crary, M. P. Obery, and D. D. Wentzloff, "Low-power wake-up receivers for resilient cellular Internet of Things," Information, vol. 16, no. 1, p. 43, 2025. https://doi.org/10.3390/info16010043

[5] Y. Liu, Z. Wang, J. Xu, C. Ouyang, X. Mu, and R. Schober, "Near-field communications: A tutorial review," IEEE Open J. Commun. Soc., vol. 4, pp. 1999-2049, 2023. https://doi.org/10.1109/OJCOMS.2023.3305583

[6] R. Kumar, V. Jain, L. W. Yie, and S. Teyarachakul, Convergence of IoT, Blockchain, and Computational Intelligence in Smart Cities. CRC Press, Taylor & Francis Group, 2024. https://doi.org/10.1201/9781003353034

[7] K. Boutiba and A. Ksentini, "On using deep reinforcement learning to balance power consumption and latency in 5G NR," in Proc. ICC, 2023, pp. 6218-6223. https://doi.org/10.1109/ICC45041.2023.10279727

[8] M. Bordin, A. Zanella, M. Zorzi, A. Testolin, and M. Polese, "Design and evaluation of deep reinforcement learning for energy saving in open RAN," in Proc. IEEE CCNC, 2025, pp. 1-6. https://doi.org/10.1109/CCNC54725.2025.10976108

[9] N. El Zarif, M. A. Hemmat, T. Dupuis, J.-P. David, and Y. Savaria, "Polara-Keras2c: Supporting vectorized AI models on RISC-V edge devices," IEEE Access, 2024. https://doi.org/10.1109/ACCESS.2024.3498462

[10] W. Wu, X. Wang, A. Hawbani, L. Yuan, and W. Gong, "A survey on ambient backscatter communications: Principles, systems, applications, and challenges," Comput. Netw., vol. 216, p. 109235, 2022. https://doi.org/10.1016/j.comnet.2022.109235

[11] M. U. Sheikh, B. Xie, K. Ruttik, H. Yiğitler, R. Jäntti, and J. Hämäläinen, "Ultra-low-power wide range backscatter communication using cellular generated carrier," Sensors, vol. 21, no. 8, p. 2663, 2021. https://doi.org/10.3390/s21082663

[12] C. A. Alabi, A. L. Imoize, M. A. Giwa, N. Faruk, and S. T. Tersoo, "Artificial intelligence in spectrum management: Policy and regulatory considerations," in Proc. ICMEAS, vol. 1, 2023, pp. 1-6. https://doi.org/10.1109/ICMEAS58693.2023.10379314

[13] D. K. McCormick, 802.11ba Battery Life Improvement-IEEE Technology Report on Wake-Up Radio, pp. 1-56, 2017.

[14] M. C. Caballé, A. C. Augé, E. Lopez-Aguilera, E. Garcia-Villegas, I. Demirkol, and J. P. Aspas, "An alternative to IEEE 802.11ba: Wake-up radio with legacy IEEE 802.11 transmitters," IEEE Access, vol. 7, pp. 48068-48086, 2019. https://doi.org/10.1109/ACCESS.2019.2909847

[15] A. Höglund, M. Mozaffari, Y. Yang, G. Moschetti, K. Kittichokechai, and R. Nory, "3GPP Release 18 wake-up receiver: Feature overview and evaluations," IEEE

Commun. Standards Mag., vol. 8, no. 3, pp. 10-16, 2024. https://doi.org/10.1109/MCOMSTD.0001.2400002

[16] W. Chen and P. Jain, "3GPP Release 18 overview: A world of 5G-Advanced," ATIS Org., vol. 2, 2023.

[17] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, and D. Brunelli, "Ultra low power wake-up radios: A hardware and networking survey," IEEE Commun. Surveys Tuts., vol. 19, no. 4, pp. 2117-2157, 2017. https://doi.org/10.1109/COMST.2017.2728092

[18] H. Bello, Z. Xiaoping, R. Nordin, and J. Xin, "Advances and opportunities in passive wake-up radios with wireless energy harvesting for IoT applications," Sensors, vol. 19, no. 14, p. 3078, 2019. https://doi.org/10.3390/s19143078

[19] R. Fromm, O. Kanoun, and F. Derbel, "An improved wake-up receiver based on the optimization of low-frequency pattern matchers," Sensors, vol. 23, no. 19, p. 8188, 2023. https://doi.org/10.3390/s23198188

[20] S. Basagni, F. Ceccarelli, C. Petrioli, N. Raman, and A. V. Sheshashayee, "Wake-up radio ranges: A performance study," in Proc. IEEE WCNC, 2019, pp. 1-6. https://doi.org/10.1109/WCNC.2019.8885974

[21] S. Mahlknecht and M. S. Durante, "WUR-MAC: Energy efficient wakeup receiver-based MAC protocol," IFAC Proc. Vol., vol. 42, no. 3, pp. 79-83, 2009. https://doi.org/10.3182/20090520-3-KR-3006.00012

[22] V. R. K. Ramachandran, E. D. Ayele, N. Meratnia, and P. J. Havinga, "Potential of wake-up radio-based MAC protocols for implantable body sensor networks (IBSN)-A survey," Sensors, vol. 16, no. 12, p. 2012, 2016. https://doi.org/10.3390/s16122012

[23] X. Wang, Y. Zhang, L. Chen, J. Liu, H. Huang, and T. Zhang, "A task-driven design approach for 6G AI-native architecture," Engineering, 2025. https://doi.org/10.1016/j.eng.2025.09.005

[24] F. Zhu, L. Chen, Y. Wang, H. Zhang, Z. Li, and S. Xu, "Wireless large AI model: Shaping the AI-native future of 6G and beyond," arXiv preprint arXiv:2504.14653, 2025.

[25] B. Li, T. Liu, W. Wang, C. Zhao, and S. Wang, "Agent-as-a-Service: An AI-native edge computing framework for 6G networks," IEEE Netw., 2024. https://doi.org/10.1109/MNET.2024.3520987

[26] H. A. Mutar, B. Duraković, A. A. Almisreb, J. Šutković, and I. Svyd, "Investigation of AI with OpenCV-Python for detecting diabetes," in Recent Trends and Applications of Soft Computing in Engineering (RTASCE), Sarajevo, Cham, 2025, pp. 217-231. https://doi.org/10.1007/978-3-031-82881-2_14

[27] A. H. M. Alaidi, Z. A. Ramadhan, J. Alrubaye, H. Mutar, and I. Svyd, "AI-based monkeypox detection model using Raspberry Pi 5 AI Kit," Sustain. Eng. Innov., vol. 7, no. 1, pp. 1-14, 2025. https://doi.org/10.37868/sei.v7i1.id393

[28] W. Saad, C. Chaccour, C. K. Thomas, and M. Debbah, Foundations of Semantic Communication Networks. John Wiley & Sons, 2024. https://doi.org/10.1002/9781394247912

[29] N. Li, Y. Zhao, H. Zhang, L. Chen, X. Wang, and T. Zhang, "Towards AI-native RAN: An operator's perspective of 6G Day 1 standardization," arXiv preprint arXiv:2507.08403, 2025.

[30] X. You, Y. Huang, C. Zhang, J. Wang, H. Yin, and H. Wu, "When AI meets sustainable 6G," Sci. China Inf. Sci., vol. 68, no. 1, p. 110301, 2025. https://doi.org/10.1007/s11432-024-4257-6