Enhancing LoRaWAN Communication for Mobile Nodes with Techniques for Predicting Signal Strength

Hristijan Slavkoski[1,2], Simeon Trendov[1], Eduard Siemens[1] and Marija Kalendar[2]
[1]*Department of Electrical, Mechanical and Industrial Engineering, Anhalt University of Applied Sciences, Bernburger Str. 55, Köthen, Germany*
[2]*Faculty of Electrical Engineering and Information Technologies, "SS. Cyril and Methodius University" in Skopje, Rugjer Boshkovik Str. 18, Skopje, N. Macedonia*
hristijan.slavkoski@student.hs-anhalt.de, {simeon.trendov, eduard.siemens}@hs-anhalt.de, marijaka@feit.ukim.edu.mk

Keywords: Adaptive control, Adaptive Data Rate (ADR), Coding Rate (CR), Energy efficiency, Internet of Things (IoT), Kalman filter, LoRaWAN, Mobile communication, Received Signal Strength Indicator (RSSI), Signal-to-Noise Ratio (SNR), Signal strength prediction, Spreading Factor (SF), Transmit Power (TP)

Abstract: The Internet of Things (IoT) has enabled a wide range of applications that depend on efficient and reliable communication between devices, even in remote or mobile scenarios. However, LoRaWAN communication faces significant challenges when nodes are mobile. This study investigates methods for predicting and improving the reliability and energy efficiency of mobile LoRaWAN communication. The Kalman filter, a lightweight yet robust algorithm, is applied to smooth noisy signal measurements and enhance decision-making. Field experiments in urban, rural, park, and free-field environments demonstrate that predictive filtering can effectively stabilize the highly variable RSSI and SNR signals typical of mobile devices, providing a more dependable basis for transmission parameter control. The performance is benchmarked against a naive reactive control strategy and further contextualized within the framework of standard LoRaWAN Adaptive Data Rate (ADR) mechanisms. Results indicate that traditional ADR, designed for stationary devices, remains overly conservative in mobile situations, maintaining higher transmission power than necessary. By contrast, predictive filtering achieves tangible energy savings without compromising reliability. This work represents a meaningful step toward more resilient and efficient LoRaWAN systems capable of supporting the mobile and dynamic applications of tomorrow's IoT landscape.

1 INTRODUCTION

The Internet of Things (IoT) has introduced a wide range of applications that rely on efficient and reliable communication between devices, even in remote or mobile scenarios. Among the many wireless communication protocols developed for IoT, LoRaWAN stands out due to its low power consumption, long-range coverage, and suitability for devices with limited processing capabilities. LoRaWAN is particularly advantageous in rural, industrial, or infrastructure-scarce environments where mobile or battery-operated devices must function independently over extended periods.

Despite its benefits, LoRaWAN communication faces significant challenges when nodes are mobile. Unlike fixed-position nodes, mobile nodes experience rapid and unpredictable fluctuations in signal strength due to factors such as distance from the gateway, environmental interference, obstacles, and movement speed. These variations can lead to packet loss, increased latency, and inconsistent connectivity, which undermine the reliability of the entire IoT system. Addressing this issue is crucial for enabling real-time mobile applications such as asset tracking, smart transportation, and environmental monitoring.

This project explores methods for predicting signal strength in mobile LoRaWAN nodes as a means of improving communication performance. By anticipating signal quality in advance, it becomes possible to adjust transmission parameters or make proactive decisions, such as delaying non-urgent transmissions until conditions improve. The core idea is to blend LoRaWAN's existing capabilities with lightweight predictive techniques that can run on simple microcontrollers and single-board computers like Arduino and Raspberry Pi.

In the following sections, the hardware setup, system architecture, and signal strength prediction techniques are presented. The results of preliminary experiments with mobile nodes are discussed, followed by an evaluation of the approach and suggestions for future improvements.

2 SYSTEM DESIGN AND IMPLEMENTATION

To investigate methods for predicting and improving LoRaWAN signal strength in mobile scenarios, a custom IoT system was developed. This section describes the architecture, components, and setup used to establish LoRaWAN communication between a mobile sender node and a stationary receiver. The goal was to create a lightweight and portable system capable of capturing signal strength data in real time as the sender moves through different environments.

This system consists of two main components: a mobile sender node and a stationary receiver gateway. The sender represents a real-world IoT device (e.g., a sensor on a delivery vehicle or a wildlife tracker), while the receiver acts as a gateway to collect data and adapt communication parameters dynamically.

2.1 Hardware Components

The hardware architecture is designed to mimic practical IoT deployments where cost, portability, and energy efficiency are critical. Both nodes use commercially available components to ensure reproducibility and scalability.

2.1.1 Sender Node (Mobile Device)

The sender node (Figure 1) is a portable, battery-powered device built around the following components:

- Microcontroller: The Arduino Mega 2560, running at 16 MHz with 256 KB flash memory, provides sufficient computational power while maintaining energy efficiency for mobile IoT applications;
- LoRa Module: A Dragino LoRa Shield v1.4 [1], equipped with the Semtech SX1276 chip [2], enables long-range communication in the 868 MHz ISM band (Europe). It supports multiple bandwidth options (e.g., 125 kHz, 250 kHz, 500 kHz), adjustable spreading factors (SF7–SF12), and transmit power up to 20 dBm;
- Antenna: An SMA-connected 868 MHz omnidirectional antenna ensures reliable signal coverage in mobile scenarios;
- Power Source: A standard USB power bank (5V/2A output) powers the Arduino, ensuring portability during field tests.

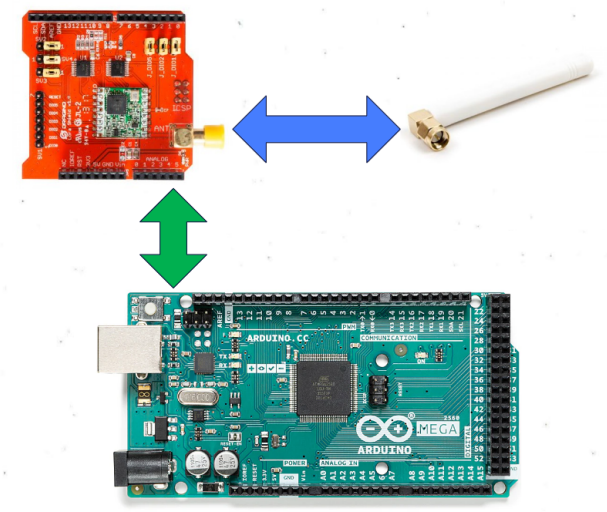

Figure 1: Mobile sender node based on Arduino Mega with Dragino LoRa Shield.

2.1.2 Receiver Node (Gateway)

The stationary gateway (Figure 2) is designed to receive data, analyze signal quality, and optimize transmission settings:

- Single-Board Computer: A Raspberry Pi 3 B+ processes incoming data. Its quad-core 1.4 GHz CPU and 1 GB RAM handle real-time filtering and adaptive algorithms;
- Dragino LoRa Bee v1.1 (SX1276 chip, 868 MHz) [3];
- Antenna: The same SMA-connected 868 MHz antenna as the sender, ensuring compatibility and omnidirectional coverage.

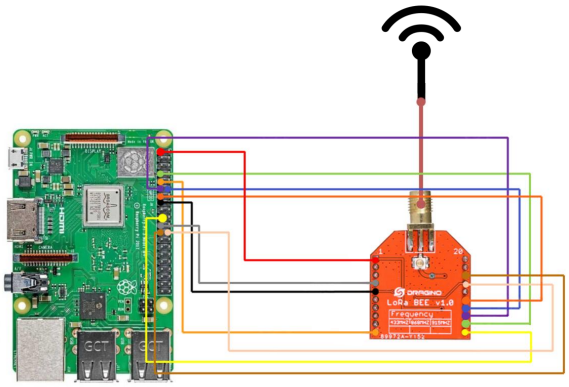

Figure 2: Stationary receiver node with Raspberry Pi 3 B+ and Dragino LoRa BEE.

2.2 Software Components

The software stack for this project consists of custom programs running on both the Arduino Mega (sender) and the Raspberry Pi 3 B+ (receiver), designed to implement adaptive LoRaWAN communication with real-time signal strength monitoring and dynamic parameter adjustment.

2.2.1 Sender (Adruino Mega)

The Arduino runs code written in C++ using the Arduino IDE and leverages the widely used LoRa and SPI libraries to manage radio communication. The sender periodically transmits numbered data packets and listens for acknowledgments (ACKs) from the receiver. Control information is carried inside these ACKs, so the sender can update its radio settings in real time without sending extra synchronization traffic. On startup, a single SYNC exchange is used to establish the initial configuration and after that, parameter updates are attached to ACKs during normal packet exchange. When an ACK contains new values for spreading factor (SF), coding rate (CR), or transmit power (TP), the sender applies them immediately. This ensures that the sender adapts to changing signal conditions, improving reliability and energy efficiency. The code includes CRC checking for packet integrity and uses timestamped logging via the serial monitor for debugging and post-processing.

Key features of the sender code include:

- Initialization: Starts with default parameters (SF=7, CR=4/5, TP=5 dBm), adjustable based on receiver feedback;
- Packet Transmission: Sends labelled packets (e.g., "PACKET: 1") and awaits ACKs, retransmitting if necessary;
- Parameter updates in ACKs: The receiver embeds SF=..., CR=..., and TP=... inside normal ACKs. The sender parses these fields and updates its configuration in place;
- Dynamic Adjustment: Increases TP if ACKs are not received, ensuring communication reliability;
- Logging: Records transmission details and parameter changes.

2.2.2 Receiver (Raspberry Pi 3 B+)

The receiver runs a Python application based on the forked pyLoRa (version 0.3.1) library, a modified implementation of the original pySX127x driver. This software interfaces with the Dragino LoRa BEE module through the SPI bus, which must be enabled via Raspberry Pi's configuration settings. The Python program continuously listens for incoming packets, logs signal quality metrics such as RSSI (Received Signal Strength Indicator) and SNR (Signal-to-Noise Ratio) and applies a simplified Kalman filter to smooth these values. Based on these filtered parameters, it dynamically adjusts the transmission power and other communication parameters by sending SYNC messages back to the sender.

The receiver's software also implements adaptive logic to increase or decrease transmission power based on threshold rules applied to RSSI and SNR values, aiming to maintain a stable link quality while minimizing power consumption. Packet delivery ratio (PDR) and CRC error rates are also monitored and logged to evaluate communication robustness. All logging data is saved with timestamps in organized files for later analysis.

Key features of the receiver code include:

- Initialization: Starts with default parameters (SF=7, CR=4/5);
- Kalman Filter: Smooths RSSI and SNR to reduce noise from mobility-induced fluctuations;
- Link Quality: Calculates Packet Delivery Ratio (PDR), average RSSI, and SNR over a sliding window;
- Adaptive Logic: Dynamically adjusts transmission parameters based on filtered RSSI and SNR thresholds. If link quality degrades, the receiver can request higher TP, a larger SF, or a more robust CR. Conversely, when conditions improve, it suggests lowering TP, reducing SF, or relaxing CR to save energy and increase throughput;
- Logging: Saves detailed metrics in timestamped files.

3 SIGNAL STRENGTH PREDICTION

This section explores the methodology used to manage and interpret signal strength data for mobile LoRaWAN nodes in an adaptive communication system. The core technique employed is the Kalman filter, a lightweight yet robust algorithm that smooths noisy signal measurements to enhance decision-making. The section breaks down the Kalman filter's mechanics, its specific application to smoothing RSSI and SNR values, its limitations, and potential avenues for future improvement [4].

Table 1: Example adaptation rules for RSSI/SNR-driven parameter changes.

Condition	Action on TP	Action on SF	Action on CR
RSSI $<$ –95 dBm or SNR $<$ –10 dB	Increase TP (up to max)	Increase SF (towards SF12)	Increase CR (towards 4/8)
–95 $\leq$ RSSI $<$ –90 dBm and SNR $<$ –5 dB	Increase TP	Keep SF	Keep CR
–90 $\leq$ RSSI $\leq$ –70 dBm, 0 $\leq$ SNR $\leq$ 5 dB	Keep TP	Keep SF	Keep CR
RSSI $>$ –70 dBm and SNR $>$ 5 dB	Decrease TP	Decrease SF (towards SF7)	Decrease CR (towards 4/5)

3.1 Understanding the Kalman Filter

3.1.1 Origins and Purpose

The Kalman filter, developed by Rudolf E. Kálmán in the 1960s, is a mathematical tool originally designed for aerospace applications, such as navigation and trajectory tracking. Its purpose is to estimate the state of a dynamic system, like the position of a spacecraft or, in our case, the signal strength of a mobile node using a series of noisy measurements. Today, its versatility has made it a staple in fields ranging from robotics to financial modelling, and it's particularly valuable in IoT systems where real-time data processing is critical [5][6].

The filter's strength lies in its ability to blend predictions (based on a system model) with measurements (from sensors), producing a refined estimate that mitigates the effects of noise. For mobile LoRaWAN nodes, this means cleaner RSSI and SNR data, which leads to more reliable adaptive decisions.

3.1.2 How It Works: The Prediction-Correction Cycle

The Kalman filter operates in two iterative steps:

- Prediction: Using a model of how the system evolves, the filter predicts the next state based on the current estimate. For signal strength, this might assume that RSSI or SNR changes gradually over time, with some random variation (termed process noise, denoted (Q)). This step also updates the uncertainty of the prediction, tracked via a covariance matrix;
- Correction: When a new measurement arrives (e.g., a raw RSSI value), the filter adjusts its prediction by incorporating this data. The measurement is assumed to include noise (quantified as measurement noise, (R)), and the filter computes a Kalman gain (K) to determine how much to trust the measurement versus the prediction. The result is an updated estimate with reduced uncertainty.

This cycle repeats with each new packet received, allowing the filter to adapt dynamically to changing conditions while smoothing out short-term fluctuations caused by the sender's mobility or environmental factors.

3.2 Applying the Kalman Filter to RSSI and SNR Smoothing

3.2.1 Why Smoothing Matters

In mobile LoRaWAN systems, RSSI and SNR measurements can fluctuate wildly due to factors like distance, obstacles, or interference. Raw data might show a sudden drop in RSSI from -80 dBm to -100 dBm due to a temporary obstruction, even if the overall signal trend remains stable. Feeding these raw values directly into the adaptive logic could trigger unnecessary adjustments, such as increasing TP when it's not truly needed.

The Kalman filter mitigates this by providing a smoothed estimate, ensuring the system responds to meaningful trends rather than transient noise.

3.2.2 Implementation Details

In this project, separate Kalman filters were applied to the RSSI and SNR data streams. Each filter was initialized with a starting estimate (e.g., -90 dBm for RSSI and 0 dB for SNR) and tuned parameters:

- Process Noise (Q): Represents the expected variation in signal strength between measurements. A small (Q) (e.g., 0.01) assumes slow changes, ideal for pedestrian-paced nodes, while a larger (Q) (e.g., 0.1) suits faster-moving nodes like vehicles;
- Measurement Noise (R): Reflects the noisiness of raw measurements, derived from observed variance in static tests (e.g., (R = 1.5)). A higher (R) reduces the filter's reliance on erratic measurements.

Tuning was tested across different mobility profiles. With slow walking, small Q and moderate R stabilized the signal without lag. With faster movement, larger Q helped track rapid drops. Packets were transmitted every 5 seconds, which is sufficient for walking but can undersample jogging. In that case increasing Q provides compensation.

Filtered outputs were then fed into the receiver's adaptive logic. Instead of reacting to every fluctuation, the controller applied hysteresis (two thresholds) and dwell time (N consecutive packets) so that only persistent conditions triggered a change. Table 1 summarizes the adaptation rules.

The controller classifies each packet using filtered RSSI and SNR. A "raise" state is declared if RSSI $<$ -92 dBm or SNR $<$ -7 dB, a "lower" state if RSSI $>$ -86 dBm and SNR $>$ -1 dB, otherwise the state is neutral. In the raise state, transmit power is increased first; if already at high power, spreading factor is suggested upward by one; if spreading factor is also at maximum, coding rate is suggested upward by one. In the lower state, transmit power is decreased first, then spreading factor or coding rate if already at the minimum power.

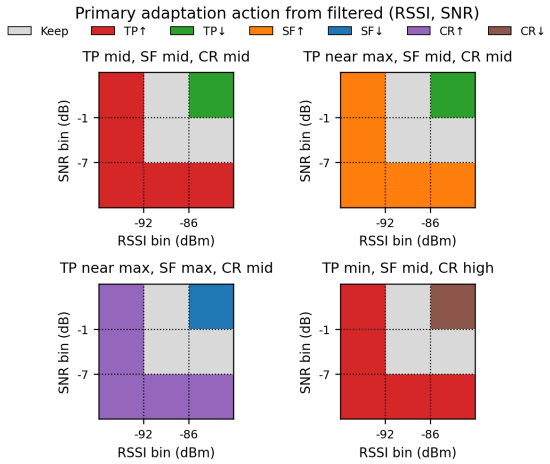

Figure 3: Decision heatmap.

Figure 3 shows this mapping on a grid of RSSI and SNR bins. Bins are value intervals; for example, the RSSI bin "-92:-86" means -92 $\leq$ RSSI $<$ -86 dBm. Hysteresis and dwell operate on top of this map to prevent brief dips from triggering changes.

4 EXPERIMENTAL PROTOCOL AND DATA COLLECTION

To evaluate the prediction and adaptation strategies under realistic conditions, a series of field experiments was conducted in urban, rural, and park environments. Rather than treating every log as equal, the focus is placed on representative runs that best capture the behavior of the system in different propagation regimes. The objective was not to maximize packet counts, but to expose the system to the full spectrum of challenges a mobile LoRaWAN node would face: interference from people and buildings in urban areas, open-line distances in rural villages, and calm baseline tests in an isolated park [7][8].

4.1 Field Environments and Motion Profiles

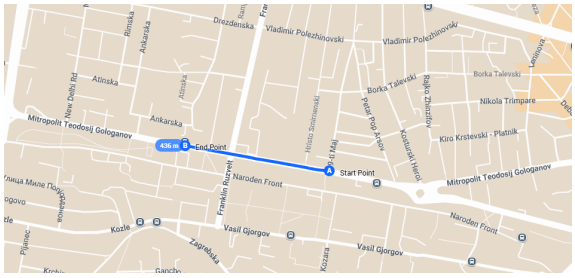

Figure 4: Urban environment.

The urban environment shown on Figure 4 was a busy street with dense pedestrian activity. The sender node was handheld and moved along sidewalks, often with people passing between sender and receiver. This created sharp, irregular fades as human bodies blocked the line-of-sight or reflected the signal. In addition, building facades introduced multipath components, making RSSI and SNR fluctuate more than distance alone would predict. This setting was deliberately chosen to stress-test the filter's ability to suppress noise without losing sight of genuine downward trends.

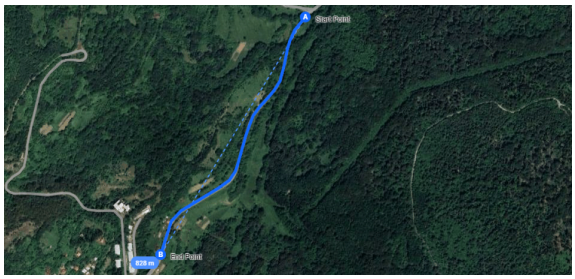

Figure 5: Rural environment.

The rural environment shown on Figure 5 was a small village route with long sightlines and sparse interference. The same sender-receiver pair was carried at a walking pace along a road with minimal obstacles, so the dominant effect was gradual distance-driven attenuation. This is the classical propagation

model scenario: path loss increases smoothly with distance, and stochastic variation is smaller than in urban conditions. This environment is particularly useful for confirming whether the Kalman filter tracks slow changes faithfully, since there is little crowd-driven volatility.

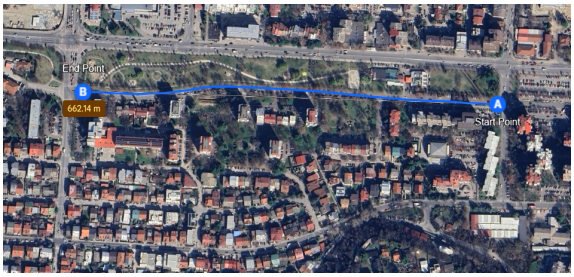

Figure 6: Park environment.

The park environment shown on Figure 6 was a large, relatively empty outdoor area with few bystanders. Here the sender could be moved at different speeds, slow walking vs. jogging, without obstructions other than occasional trees. The park provided the cleanest conditions and served as a "baseline sanity check" to see the best possible performance of the smoothing and control loop. Because interference was minimal, filtered and raw metrics could be compared in isolation to reveal how much variance the algorithm removed, independent of external disruptions.

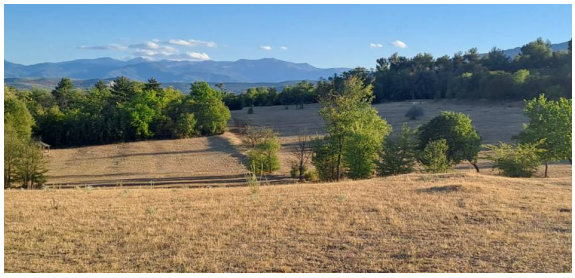

Figure 7: Free field environment.

The free-field environment shown on Figure 7 was situated on a gentle, rolling hill, providing a clear, elevated vantage point over open ground. The receiver was placed near the crest, while the sender node was carried along the slope and across the adjacent flat area. With no buildings or dense crowds, this environment was largely free of the multipath effects and sharp blockages seen in urban settings. The primary sources of signal variation were distance-driven path loss and changes in antenna orientation due to the operator's movement. This scenario served as a controlled baseline to isolate the fundamental dynamics of the wireless channel, demonstrating the filter's performance under near-ideal, line-of-sight conditions.

Across all four environments, two motion profiles were tested:

- Slow traverse $\approx$ 4–5 km/h: a casual walking pace with frequent small pauses;
- Fast traverse $\approx$ 9–10 km/h: covering the same route in tens of seconds rather than minutes.

These contrasting speeds were essential to highlight the filter's responsiveness, since the same parameters can be well-suited for one regime and suboptimal for the other.

4.2 System Roles and Packet Flow

The sender initialized with default LoRa parameters and began by transmitting a SYNC message. It then sent sequentially numbered data packets at a fixed cadence, each carrying a CRC for integrity. The receiver logged every incoming packet, computed raw RSSI and SNR, applied the Kalman filter t o both streams, and wrote the results to file. W hen thresholds were crossed, the receiver decided whether to escalate or maintain transmit power, and it annotated the log with both the raw decision (what would have happened without smoothing) and the filtered adaptation (the actual outcome). This dual record is invaluable because it allows later analysis of how often the smoother prevented unnecessary power changes.

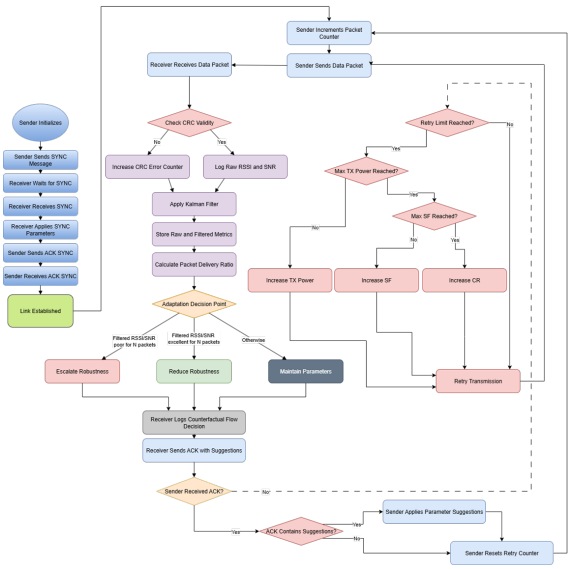

Figure 8: Adaptive packet flow and decision loop between sender and receiver.

The complete packet flow and decision logic, encompassing both the initial handshake and the main data cycle, are illustrated in Figure 8. In a typical data exchange, the sender emits a packet (e.g.,

"PACKET 1"). The receiver processes it (logging, filtering, threshold checks) and may reply with an adaptive "ACK 1" containing parameter change suggestions.

4.3 Logging and Metrics

Each log line contained a timestamp with millisecond precision, the received message label, raw and filtered RSSI (dBm), raw and filtered SNR (dB), CRC validity, and the packet delivery ratio (PDR) as observed up to that point. Adaptation decisions were also embedded in the log, for example:

```
2025-06-22 21:59:10.024 Received:
    Message: PACKET: 11;
    RSSI_raw=-105,
    RSSI_filtered=-73.8, SNR_raw=4.0,
    SNR_filtered=8.5, CRC=Valid,
    PDR=100.00%
2025-06-22 21:59:10.025 Adaptation:
    TP=0 (Filtered) | Raw decision: No
    change
2025-06-22 21:59:10.028 Sent: ACK 11
```

This made it straightforward to parse the logs and reconstruct both the "counterfactual" raw control path and the actual filtered path. To assess how mobility speed influenced estimation quality, a simple motion proxy was introduced: the median absolute derivative of filtered RSSI over time $\text{median}\left(\left|\frac{d\,\text{RSSI}_f}{dt}\right|\right)$ [dB/s]. Runs with higher values corresponded to faster movement or more abrupt environmental changes, while runs with lower values reflected slow traverses.

4.4 Representative Dataset

From the broader set of collected logs, four representative cases (urban, rural, park and free field) were selected, as they clearly illustrate different stress factors on the system. In the urban case, the logs reveal moments where people walking past triggered sharp raw RSSI drops, which the filter largely suppressed. In the rural case, the filter tracked the smooth decline well, showing minimal lag. In the park case, slow vs. fast traverses under otherwise identical conditions were directly comparable, highlighting the need for a speed-aware filter setting. The free field case offered near-ideal conditions, with clear line-of-sight and minimal interference, providing a baseline to assess filtering performance in more complex settings.

5 RESULTS AND ANALYSIS

Building on the experimental setups described in the previous section, this section evaluates how the prediction-and-control loop behaved under real mobility. Rather than re-describing the measurement environments, the focus here is on the observed behaviors and how the Kalman filter plus adaptation policy responded. Four representative traces, one urban, one rural, one park, and one free field, provide contrasting conditions: intermittent crowd-induced fading, smooth distance-driven attenuation, a clean baseline, and a near-ideal line-of-sight scenario. The results are organized around four questions:

- Reliability: was packet delivery preserved across environments despite mobility?
- Stability: did the Kalman filter reduce volatility in RSSI and SNR without masking true degradations?
- Responsiveness: how well did the adaptation logic react in slow versus fast traverses, and did smoothing sometimes introduce lag?
- Efficiency: did the system save energy by preventing unnecessary parameter increases?

To answer these, time-series comparisons of raw vs. filtered RSSI/SNR, variance-reduction summaries, and slow/fast motion panels are presented. Together these reveal where the controller saved power by ignoring transient fades, and where it risked under-reacting in rapid movement.

5.1 Reliability Outcomes

Across all representative runs, end-to-end reliability remained high. Data packets were received with valid CRCs and acknowledgments were consistently returned. In the urban trace, short fades appeared when pedestrians crossed the line-of-sight between sender and receiver, causing temporary signal dips. These are exactly the kinds of disturbances that a threshold-only controller would misinterpret as persistent degradation, leading to unnecessary power increases. The Kalman filter smoothed such events, and the adaptation logic largely avoided reacting to them, demonstrating the stabilizing value of prediction.

By contrast, the rural trace showed long, gradual declines in RSSI as distance increased, with little short-term volatility. Here the filter tracked the underlying trend closely, and the adaptation policy correctly escalated transmit power only when sustained attenuation warranted it. This validates that the system does not miss genuine link degradation even when smoothing is active.

5.2 Smoothing and Stability

To assess stability, the variability of raw and filtered metrics was compared across environments. Variability was measured in terms of variance, which reflects how much the signal fluctuates around its mean. High variance in RSSI or SNR makes it harder for a controller to distinguish between temporary dips and true deterioration, because thresholds are crossed frequently in both cases. By reducing variance, a filter allows the controller to base its decisions on longer-term structure rather than reacting to short spikes.

In both the urban and rural traces, the Kalman filter reduced RSSI fluctuations by roughly one-third and SNR fluctuations by about one-fifth. These are meaningful reductions: a one-third drop in RSSI variance means that the signal is noticeably steadier and less likely to oscillate around decision boundaries. In practical terms, many borderline threshold crossings that appear in the raw data are smoothed away in the filtered data. This prevents the system from bouncing between power levels in response to noise rather than genuine changes.

The benefit of smoothing was most visible in the urban trace, where pedestrians occasionally blocked line-of-sight. Each crossing created sharp but brief fades in the raw RSSI. Without filtering, the controller would interpret these as repeated losses, leading to unnecessary power escalations. The Kalman filter suppressed these momentary dips, ensuring that only persistent attenuation triggered action. In the rural environment, where the dominant trend was distance-driven path loss, the filter did not alter the overall trajectory but still damped the smaller oscillations caused by reflections and multipath. In the park, which served as a cleaner baseline, the differences were less dramatic, but the filter still delivered slightly steadier estimates.

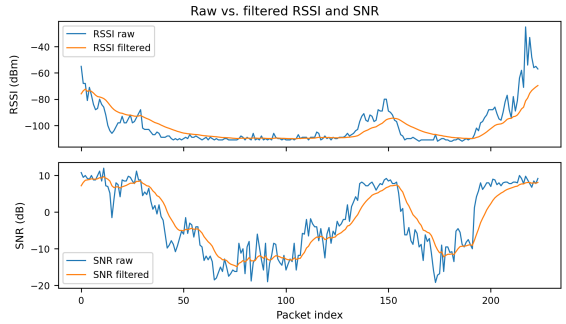

Figure 9: Time-series of raw and filtered RSSI (top) and SNR (bottom).

Time-series plots (Figure 9) capture this visually: raw RSSI curves whip in and out of thresholds, while the filtered trajectories remain more consistent.

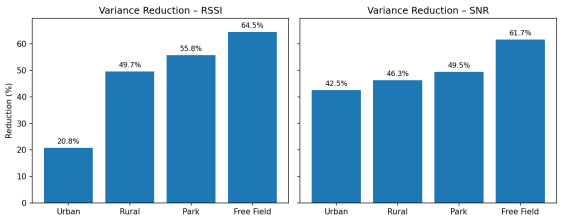

Figure 10: Variance reduction across urban, rural, park and free field environments.

Variance-reduction bar charts (Figure 10) summarize the effect across all four environments, confirming that the Kalman filter improved stability regardless of context. Importantly, the long-term trends were preserved: the filter smoothed noise but did not mask genuine degradation.

This balance is critical. An adaptive system that chases every spike will waste energy and become unstable, while one that smooths too aggressively may miss the onset of real problems. By achieving less jitter while still tracking the overall decline, the Kalman filter demonstrated that stability and responsiveness can coexist. These results show why smoothing is not just cosmetic but an essential component of reliable control in mobile LoRaWAN links.

5.3 Motion Sensitivity: Slow vs. Fast Traverses

While smoothing improves stability, its responsiveness depends on how quickly the link conditions evolve. To examine this, two traverses of similar distance but very different paces were compared. The slow traverse was a casual walk ($\approx$4–5 km/h), while the fast traverse ($\approx$9–10 km/h) covered the same route in only tens of seconds rather than minutes.

In the slow case, the filtered RSSI trajectory followed the gradual decline almost perfectly. Small fades caused by minor movements or short obstacles were suppressed without introducing noticeable delay. The adaptation logic escalated transmit power only when the decline continued over many packets, which was the intended behavior. This shows that the filter was well matched to slow mobility where trends unfold gently and there is enough time to accumulate evidence before reacting.

The fast case told a different story. Here the raw RSSI dropped sharply as distance grew. The filtered curve trailed behind because the process noise parameter Q had been set conservatively. The filter therefore underestimated how severe the path loss had become, and the controller hesitated before raising transmit power. Connection was not lost, but the delay made clear that the filter's tuning was not equally effective

across different speeds of motion.

Figure 11: Comparison of slow (top) and fast (bottom) traverses.

This highlights a trade-off in parameterization. Small Q produces smooth and stable estimates, which is ideal for slow environments, but it creates lag during rapid movement. Larger Q can follow fast changes more closely, but it makes the system sensitive to short-term noise. A practical refinement is to let Q depend on movement speed. For example, when motion is slow, the filter can remain conservative, while in faster traverses it can become more agile. Another complementary idea is to add hysteresis or dwell thresholds on top of the filter output. Hysteresis ensures that a single threshold crossing does not trigger a power escalation unless the signal stays below the threshold for several packets. Dwell counters formalize this by requiring persistence before a state change is accepted. Together, these refinements help separate temporary dips from real deterioration.

Figure 11 illustrates this contrast. The slow panel shows the filtered signal hugging the underlying trendline with minimal jitter. The fast panel shows the filtered trajectory lagging behind the sharp raw decline, which motivates the need for speed-aware tuning and simple rules like hysteresis to keep decisions both stable and responsive.

5.4 Control Decisions: Raw vs. Filtered Paths

Because the receiver logs both the raw metrics and the filtered adaptation outcomes, it is possible to compare directly what would have happened without smoothing to what actually occurred. This comparison is useful because it exposes the counterfactual decisions the system avoided.

In the urban run, the raw path would have triggered multiple increases in transmit power. Each short fade, caused by pedestrians temporarily blocking line-of-sight, would have been misinterpreted as lasting degradation. The filtered path instead remained steady through these disturbances. The network stayed reliable because acknowledgments continued to arrive, but energy was saved by not escalating power unnecessarily. This shows how filtering can reduce false alarms and improve efficiency without compromising delivery.

In the rural and park runs, the channel varied more smoothly. Here the raw and filtered decisions often aligned because both saw the same gradual trend. When the filtered RSSI stayed below the threshold for a sustained window of time, the controller did raise transmit power. This confirms that the filtering process did not ignore genuine degradation. Instead, it acted like a safeguard, requiring stronger evidence before changing state.

The key insight is that smoothing reshapes the decision landscape. Without filtering, the controller reacts to every fluctuation, which means wasted energy and constant toggling of transmit power. With filtering, the controller becomes more selective, changing state only when the evidence is both strong and persistent. In practice, this behavior can be sharpened further with dwell counters or hysteresis rules that force the system to wait through several consecutive low values before escalating power. Such mechanisms extend the principle already demonstrated here: save energy by ignoring short dips, but never miss a true decline.

This counterfactual view provides one of the clearest demonstrations of the filter's value. The logs show that the filtered adaptation path is not only smoother in appearance but also more rational in its decisions. It suppresses impulsive reactions while still responding decisively to genuine signal degradation, striking a balance between reliability and efficiency that a raw threshold-based approach cannot achieve,

5.5 Energy Impact and ADR Comparison

A central goal of integrating predictive filtering is to enhance the reliability of mobile LoRaWAN links while simultaneously optimizing energy efficiency. This section provides a quantitative evaluation of the energy savings achieved by the Kalman filter-based adaptive controller. The performance is benchmarked against a naive, reactive control strategy and contextualized within the framework of standard LoRaWAN Adaptive Data Rate (ADR) mechanisms.

5.5.1 Methodology for Energy Measurement

To ensure a fair and accurate assessment, a transparent calculation methodology was developed, rooted in

the physical properties of radio communication and the empirical data gathered during experiments. This approach is generalizable to any adaptive LoRa system.

The first step was to establish a baseline for the energy cost of transmission decisions. The Semtech LoRa Calculator [9] was used as an authoritative source to determine the fundamental physical layer costs for our specific packet configuration. For any given set of transmission parameters (e.g., Spreading Factor, Bandwidth, Coding Rate, Transmit Power), the calculator provides two key deterministic values:

- Time on Air (ToA): The precise duration required to physically transmit a single packet;
- Transmit Current (I_tx): The current consumption of the radio transceiver for a given output power level.

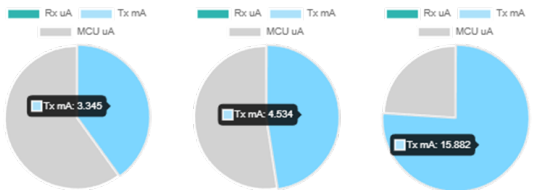

Figure 12: Examples from LoRa Calculator.

These values are intrinsic to the radio hardware and modulation settings, allowing for the creation of a definitive energy cost model. The energy consumed by any single packet transmission can therefore be calculated as a function of its specific parameters. Some of the results that were retrieved are shown on Figure 12.

The second step was to extract the two decision paths from the logs:

- The Implemented Adaptive Path: The actual sequence of parameter settings (e.g., TP=0 in the log example) commanded by our system based on the Kalman-filtered RSSI and SNR values;
- The Logged Reactive Path: The alternative decision (Raw decision: No change in the example) that was calculated instantaneously using the raw, unfiltered signal metrics and the same threshold logic. This path represents a baseline controller that reacts to every sample without any noise suppression.

The total energy consumption for each path was then computed by summing the cost of every individual packet according to the predefined energy model. The overall effectiveness of the adaptive controller is expressed by the percentage of energy saved compared to the reactive baseline that was logged alongside it:

$$E_{\text{saved}}(\%) = 100\left(1 - \frac{E_{\text{adaptive}}}{E_{\text{reactive}}}\right). \quad (1)$$

This methodology provides a robust, direct comparison based on empirical data captured during operation, isolating the energy impact of the intelligent control algorithm.

5.5.2 Quantified Savings and Strategic Comparison

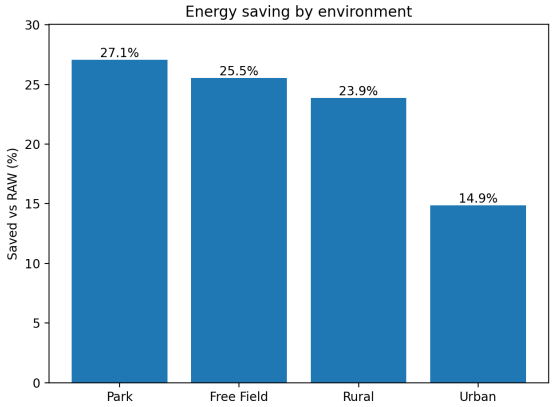

Figure 13: Energy saving by environment.

Applying this analytical framework to the collected datasets from all four environments yielded the results summarized in Figure 13.

The results demonstrate that the value of prediction is context-dependent. In the Park environment, characterized by dense vegetation and frequent short-duration fades, the adaptive controller achieved the largest reduction, lowering transmission energy by 27.1%. The high variability of signal conditions in this setting made the baseline approach prone to over-reacting to transient dips, while predictive filtering provided stability by smoothing short-term fluctuations.

In the Free-field scenario, with unobstructed line-of-sight conditions, the energy saving was 25.5%. Here, the main disturbances arose from self-induced variation, such as changes in device orientation during movement. The filtering method successfully distinguished these minor fluctuations from true link degradation, preventing unnecessary parameter escalation.

The Rural environment delivered a still substantial saving of 23.9%. Although rural channels are generally smoother, sporadic shadowing and occasional non-line-of-sight segments caused the baseline method to hold higher transmission power for longer than necessary. Predictive filtering avoided this by reducing power once the temporary disturbances passed.

Finally, the Urban environment, with its dense multipath and micro-blockages, showed the smallest but still meaningful saving of 14.9%. Here, signal conditions were highly dynamic but often recovered quickly, limiting the advantage of filtering. Nonetheless, the adaptive strategy still reduced unnecessary overhead, demonstrating consistent benefit across all tested environments.

To further contextualize these results, a comparison was made with the principles of a standard LoRaWAN ADR strategy. ADR is designed for quasi-static end devices and optimizes for long-term stability, often leading to conservative parameter settings that are slow to scale down after a degradation event [10].

In contrast, the adaptive strategy evaluated here is designed for mobility. It requires sustained evidence of a poor link before escalating parameters and, crucially, is quick to return to more efficient settings once conditions improve. This agility ensures that the system does not waste energy by remaining at conservative settings longer than necessary. Therefore, while both strategies aim for reliability, the proposed approach achieves it with a lower energy footprint in dynamic mobile scenarios [11].

In conclusion, the integration of predictive filtering fundamentally changes the control paradigm from a reactive to a proactive one. By making evidence-based decisions that distinguish between noise and genuine degradation, the system delivers tangible energy savings without sacrificing c ommunication integrity, a critical advancement for power-constrained mobile IoT applications.

6 DISCUSSION AND FUTURE WORK

The analysis above demonstrates that prediction and smoothing techniques significantly improve the stability of a mobile LoRaWAN link, especially under conditions of fading and mobility. Yet, as in any real system, the story does not end with promising results: there are caveats that must be acknowledged, and there are opportunities to refine the design further. This section steps back from the raw numbers to reflect on what they mean in practice. It considers both the limitations of the current setup and the directions in which the approach can evolve. The aim is not only to highlight what worked, but also to provide a realistic picture of where improvements are needed and how they might be achieved in future work

6.1 Limitations of the Current Approach

Every experimental system sits at the intersection of idealized models and messy reality. The current work is no exception, and several limitations are worth discussing.

First, the effectiveness of the Kalman filter is highly dependent on parameter tuning, particularly the process noise Q and the measurement noise R. In our experiments, a conservative Q produced excellent stability during slow movement, where raw RSSI and SNR fluctuate gently. However, when the same parameters were applied to fast traverses, the filter lagged behind sudden changes in distance, briefly underestimating the severity of path loss. This shows that a single fixed set of parameters may not be universally optimal. The system worked well in many cases, but its responsiveness was not equally strong across all mobility profiles.

Second, while the dataset covers four distinct environments (urban, rural, park, and free field), there are still gaps. Urban experiments focused on pedestrian interference, but did not include scenarios such as vehicles moving at higher speeds, indoor propagation through walls, or transitions between line-of-sight and non-line-of-sight. Similarly, rural tests were limited to village roads with relatively flat terrain; mountainous or forested paths could reveal different attenuation characteristics. Thus, while the results are representative, they are not exhaustive.

Third, the analysis relies on a specific sender–receiver pair (Arduino Mega and Raspberry Pi 3 B+ with SX127x modules). These choices ensured control and reproducibility, but they may not capture the full variability of real deployments, where different transceivers, antennas, or firmware stacks could alter performance. Packet delivery ratios were high in our runs, but this was partly due to the controlled setup and fixed cadence of transmissions. In a denser or noisier spectrum, with many concurrent nodes, reliability could drop and interact differently with the filtering logic.

Finally, a fundamental limitation lies in defining the optimization goal itself. The current system prioritizes reliability above all else, minimizing power only when the link is exceptionally good. However, a real-world deployment might need to balance this against other constraints, such as maximizing battery life even at the cost of occasional packet loss, or minimizing network-wide interference. The adaptation logic presented here is a single strategy, and exploring different cost functions for different applications remains an open challenge.

Taken together, these limitations frame the system as a prototype and proof-of-concept. It demonstrates that filtering stabilizes decisions and prevents overreaction, but it does not yet provide a fully general solution that covers all environments and devices.

6.2 Future Research Directions

The results open several promising directions for future work. One natural enhancement would be to integrate additional sensing modalities, such as GPS or inertial sensors, alongside the radio link metrics. By combining location and movement data with filtered RSSI and SNR, the system could better anticipate when signal degradation is likely to occur, for example at the edge of coverage or when turning behind an obstruction. A GPS-assisted controller could also enable spatially aware adaptation policies, learning how specific routes or areas typically affect link quality and adjusting parameters preemptively.

Another direction is to test the framework in more diverse mobility patterns and environments. The present study focused on pedestrian mobility in urban, rural, and park conditions, but many IoT deployments involve vehicles, drones, or mixed indoor–outdoor transitions. Each of these contexts presents different propagation challenges, from fast Doppler-induced fluctuations in vehicular links to sudden attenuation indoors. Extending the experiments to such settings would demonstrate whether filtering remains robust under higher dynamics and more abrupt changes in channel conditions.

A further opportunity lies in integration with machine learning methods [12]. Threshold-based decision logic provides a solid baseline, but predictive models could extract patterns from historical data and anticipate link quality changes before they occur. Reinforcement learning could, for example, optimize parameter updates to maximize long-term delivery ratios, while classification models could identify mobility contexts, such as walking versus driving, and adjust adaptation strategies accordingly. These approaches would preserve the stabilizing role of filtering while making control more proactive and context-aware.

Finally, scaling to multi-node scenarios represents an essential step toward real-world applicability [13]. LoRaWAN networks are typically shared by many devices, each competing for airtime. Evaluating the system in settings with multiple transmitters would highlight how filtering-based adaptation interacts with interference, collisions, and duty-cycle limits. This would also create opportunities to investigate fairness, ensuring that one device's adaptive gains do not reduce performance for others.

Taken together, these directions show a clear path forward: enrich adaptation with additional sensing, broaden validation across more demanding mobility patterns, explore predictive learning methods, and expand evaluation to larger, multi-node networks. Each of these steps would build on the central insight demonstrated here: smoothing noisy link metrics before making adaptation decisions provides a more stable foundation for reliable mobile LoRa communication.

7 CONCLUSIONS

This project has demonstrated that predictive filtering and adaptive control can significantly enhance the reliability and energy efficiency o f m obile LoRaWAN communication. Through systematic field experiments in urban, rural, park and free field environments, we have established that even lightweight prediction methods such as the Kalman filter c an effectively stabilize the unpredictable RSSI and SNR signals typical of mobile devices, creating a more dependable foundation for transmission parameter control.

The analysis revealed that the benefits of filtering are not uniform across all situations. During slow movement, the filter c losely f ollowed g radual signal trends, while faster motion revealed a slight lag that indicates the need for adaptive tuning of filter parameters based on mobility patterns. This comparison between raw and filtered d ecision p aths c onfirmed that filtering provides tangible benefits for both energy efficiency a nd l ink s tability, n ot m erely c osmetic improvements.

The most important results show clear improvements in signal stability, with the filter reducing RSSI variations by approximately one-third in challenging urban conditions. This stabilization translated into measurable energy savings of 14.9% (Urban), 27.1% (Park), 23.9% (Rural), and 25.5% (Free field) by preventing unnecessary power increases during short-term signal problems. Importantly, these efficiency gains were achieved without compromising reliability, as the system maintained high packet delivery rates across all test scenarios.

The analysis clearly shows what would occur without the filter: a basic controller would frequently overreact to temporary signal changes, causing unnecessary power spikes and energy waste. This is particularly evident in urban environments where pedestrian movement causes brief signal drops that trigger false alarms in unfiltered s ystems. T he compar-

ison with standard ADR methods further supports our approach, demonstrating that traditional systems designed for stationary devices remain overly conservative in mobile situations, maintaining higher transmission power than necessary for extended periods.

Beyond these technical insights, the work highlights the value of real-world experimentation. Testing in urban, rural, and park conditions provided a comprehensive understanding of system performance that simulations alone could not reveal. While the current approach has limitations, particularly in its threshold-based logic and single-device testing, it establishes a solid foundation for future development with enhanced sensing capabilities, diverse mobility patterns, and multi-node network configurations.

Nevertheless, this research provides convincing evidence that intelligent signal processing can directly extend battery life without reducing communication quality. The project confirms that filtering represents not just an optional enhancement but an essential component for effective mobile LoRaWAN systems, transforming unpredictable radio signals into a trustworthy basis for autonomous decision-making and sustainable operation. This work represents a meaningful step toward more resilient and efficient LoRa systems capable of supporting the mobile and dynamic applications of tomorrow's IoT landscape.

REFERENCES

[1] Dragino, "Lora shield: Long range wireless transceiver for arduino - datasheet," 2016.

[2] Semtech Corporation, "Sx1276/77/78/79 – 137 mhz to 1020 mhz low power long range transceiver," Aug. 2016.

[3] Dragino, "Lora gps hat - user manual," Mar. 2019.

[4] I. Aqeel, E. Iorkyase, and H. Zangoti, "Lorawan-implemented node localisation based on received signal strength indicator," vol. 13, 2022, pp. 117–132, doi: 10.1049/wss2.12039.

[5] H. Vo, V. H. L. Nguyen, and V. L. Tran, "Advance path loss model for distance estimation using lorawan network's received signal strength indicator (rssi)," vol. 12, 2024, pp. 83 205–83 216, doi: 10.1109/ACCESS.2024.3412849.

[6] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," vol. 21, no. 3, 2019, pp. 2568–2599, doi: 10.1109/COMST.2019.2911558.

[7] F. Lemic, A. Behboodi, and J. Famaey, "Location-based discovery and vertical handover in heterogeneous low-power wide-area networks," vol. 6, no. 6, Dec. 2019, pp. 10 150–10 165, doi: 10.1109/JIOT.2019.2935804.

[8] S. Trendov, B. Zadoienko, and E. Siemens, "Evaluation of lorawan's static and dynamic capabilities and its limitations for iot applications," in Proc. 11th Int. Conf. Applied Innovations in IT (ICAIIT), Mar. 2023, pp. 7–16, doi: 10.25673/101901.

[9] Semtech Corporation, "Lora modem designer's guide," no. AN1200.13, 2019.

[10] S. Trendov, M. Gering, and E. Siemens, "Impact of lorawan transceiver on end device battery life-time," in 2023 30th International Conference on Sys-tems, Signals and Image Processing (IWSSIP), Ohrid, North Macedonia, 2023, pp. 1–5, doi: 10.1109/IWSSIP58668.2023.10180293.

[11] A. Augustin, J. Yi, and T. Clausen, "A study of lora: Long range & low power networks for the internet of things," vol. 16, no. 9, Sep. 2016, p. 1466, doi: 10.3390/s16091466.

[12] K. Z. Islam, D. Murray, and D. Diepeveen, "Ma-chine learning-based lora localisation using multiple received signal features," vol. 13, no. 4, Jun. 2023, pp. 133–150, doi: 10.1049/wss2.12063.

[13] M. Jouhari, N. Saeed, and M.-S. Alouini, "A survey on scalable lorawan for massive iot: Recent advances, potentials, and challenges," vol. 25, no. 3, 2023, pp. 1841–1876, doi: 10.1109/COMST.2023.3274934.