

Data Processing with Predictions in LoRaWAN

Mariusz Nowak ^{*} , Rafał Różycki , Grzegorz Waligóra , Joanna Szewczyk , Adrian Sobiesierski and Grzegorz Sot

Institute of Computing Science, Poznan University of Technology, 60-965 Poznan, Poland

* Correspondence: mariusz.nowak@cs.put.poznan.pl; Tel.: +48-61-665-2921

Abstract: In this paper, the potential to reduce the energy consumption of end devices operating in a LoRaWAN (long-range wide-area network) is studied. An increasing number of IoT components communicating over wireless networks are powered by external sources. Designers of communication systems are concerned with extending the operating time of IoT, hence the need to look for effective methods to reduce power consumption. This article proposes two algorithms to reduce the energy consumption of end devices. The first algorithm is based on the use of a measured value prediction, and the second algorithm optimizes the antenna gain of the end device. Both algorithms have been implemented and tested. The test experiments for reducing energy consumption were conducted independently for the cases with the first algorithm and then for the second algorithm. The possibilities of reducing energy consumption were also investigated for the case when both algorithms work together. The proposed predictive algorithm reduced energy consumption the least. Better results in reducing energy consumption were guaranteed by the algorithm optimizing antenna power. The greatest gain was achieved using both algorithms simultaneously. Tests of the developed algorithms, in laboratory conditions and in conditions with a change in the distance between the end device and the LoRa gateway, confirmed the possibility of reducing energy consumption during the transmission of measurement data in a low-energy wireless LoRaWAN. Reducing electric energy consumption by even a few percent for a single device can result in significant savings on a global scale.

Keywords: LoRa; LPWAN; LoRaWAN; Internet of Things; energy consumption; energy optimization



Citation: Nowak, M.; Różycki, R.; Waligóra, G.; Szewczyk, J.; Sobiesierski, A.; Sot, G. Data Processing with Predictions in LoRaWAN. *Energies* **2023**, *16*, 411. <https://doi.org/10.3390/en16010411>

Academic Editor: Jose A. Afonso

Received: 8 November 2022

Revised: 20 December 2022

Accepted: 23 December 2022

Published: 29 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The extensive research progress observed in recent years in the field of the Internet of Things (IoT) involves the development of effective and energy-efficient communication channels—low-power wide-area networks (LPWAN)—and the field of low-power wireless networks is rapidly expanding. Edge devices operating in IoT communication networks perform their tasks both in heavily urbanized areas and in the field, where their access for service purposes may prove difficult. Therefore, edge devices are required to faultlessly operate for long periods of time, without human intervention in the form of servicing and replacement with an external power source [1]. It is assumed that to ensure the efficient and reliable operation of edge devices in low-energy networks, security and energy consumption issues must be optimized [2–4]. Five parameters for reliable and secure transmission should be configured as standard in each edge device operating in a wireless network: spreading factor (SF), carrier frequency (CF), bandwidth (BW), coding rate (CR), and transmission power (TP) [4]. By selecting the optimal value for each of the five transmission parameters, it is possible to reduce the energy consumption of edge devices during their operation in a low-energy network. Edge devices operating in IoT networks usually perform such tasks as measuring the value of a specific parameter, processing the measured values locally, and sending the processed measurement data to the gateway within a specified time [5,6]. The performance of these tasks by edge devices consumes a certain amount of

electricity. To estimate the total operating time of an edge device powered by an external energy source, it is necessary to know the exact amount of energy consumption during each stage of its normal operation [7,8]. The low-power wireless LoRaWAN protocol used for communication, using standard implementations of LoRa (long range) communication technology, has a low power consumption and allows data to be transmitted over long distances [9]. Using selected methods of optimizing energy consumption, often based on elements of artificial intelligence, it is possible to maximize the operating time of an edge device operating in a LoRaWAN [10–12]. This article presents proposals for optimizing the energy consumption of an edge device operating in a LoRaWAN using an algorithm that predicts the measured values of a selected parameter and by appropriately adjusting the antenna gain as a function of the received signal parameters. The optimization algorithms proposed by the authors extend the operating time of the edge device and reduce the cost of its operation. There are thousands of edge devices operating in IoT networks. Reducing the operating cost of each IoT will result in significant savings for the operation of the entire network, which is especially important in today's economic climate, where energy and electronic components are becoming increasingly expensive.

This paper is organized as follows. In Section 2, related research and contributions are described. In Section 3, LoRa wireless communication technology and a LoRaWAN wireless communication protocol for IoT applications are characterized. The proposal of the test apparatus for measuring the energy consumption of end devices operating in LoRaWAN and optimization algorithms is presented in Section 4. Tests of the developed algorithms for optimizing antenna gain and predicting the temperature value are placed in Section 5. Section 6 presents our concluding remarks.

2. Related Works

LoRa communication technology and the LoRaWAN protocol have recently been introduced in IoT applications [13]. These are some of the most effective wireless communication systems with a long range and low power consumption. One review article [14] compares different communication solutions, pointing out the benefits of the LoRaWAN protocol. The LoRaWAN communication protocol is immune to interference. In the article [15], the authors tested a range of LoRaWANs by conducting signal tests indoors and outdoors. They initially assumed a significant loss of signal quality during indoor measurements, but the data were lost due to the signal modulation implemented in the technology. The researchers also tested the signal-to-noise ratio (SNR) in crowded areas where a significant number of devices could interfere with the transmitted signal. The tests showed a moderate impact of interference on signal quality, with all sent packets being received by the end device. Regardless of the chosen type of communication in IoT systems (Bluetooth, ZigBee, GSM (global system for mobile communications), WiFi (wireless fidelity)), one of the most important aspects is energy consumption. In the study of [16], the authors addressed the issue of power management. Their goal was to reduce the power consumption of devices using LoRaWAN by reducing the number of transmission channels used and reducing the SF (spreading factors). The determination of the indicated parameters was made by applying machine learning with reinforcement aiming to minimize energy consumption. It is assumed that minimization effects do not reduce the number of packets received by end devices. To this end, the signal-to-noise ratio occurring during data transmission was monitored. The authors made optimizations with the aim of connecting power devices to LoRaWAN using renewable energy, while trying to reduce the system's dependence on power from the grid as much as possible. Reducing energy consumption by reducing signal power can lead to a drastic reduction in network coverage and, as a result, the loss of packets transmitted to devices that are not covered in the network. The analysis of data loss caused by the modification of network parameters was undertaken by the authors of [17], which verified the value of SNR depending on the adopted SF and BW. Their study showed that the signal-to-noise ratio increases with increasing transmitter power, while the transmission rate decreases. The number of transmitted packets is also influenced by

an increase in bandwidth, which then increases the data rate. Because of the considerable distance that separates the devices in LoRaWAN, this technology is equipped with an error correction technique called forward error correction (FEC). This technique entails sending multiplexed data packets, making it possible to inspect the received data. LoRa can be used to control errors; however, this is associated with a loss of transmission speed on each bandwidth. LoRa technology has piqued the interest of many researchers, as evidenced by numerous articles describing the potential for improved performance. The authors of [18] explored a problem concerning data collisions when there is a large accumulation of end devices connected to a LoRaWAN network. The described situation results in a decrease in data transmission rate or even loss of transmitted packets. The solution proposed by the authors assumes a modification of the MAC protocol (medium access control) based on changes made in the receiver, which will make it possible to simultaneously receive packets, even considering odd SF values. Compared to the current solution, performed simulations of operation using the modified MAC protocol show increased data transmission performance for a network containing several thousand devices [19]. This involves reducing the number of data collisions during transmission. However, low power consumption in the communication and signal range cannot be the only solution to the challenges of IoT communication. An equally important aspect is the security of the communication protocol. One of the greatest challenges with radio communication is that the radio signal itself can be intercepted relatively easily. LoRaWANs, due to their low-energy nature, can be used in a wide variety of systems, including those that can be targeted by various types of attacks. Therefore, it is important to ensure the confidentiality of transmission and confirm the identity of the sender to protect against man-in-the-middle attacks and control the integrity of messages. Over the past few years, a variety of solutions have been suggested to enable IoT devices to communicate securely over a LoRaWAN. Articles that address security aspects include [20], which proposes that one of the factors that can confirm the identity of the sender of a message is location. In addition, research focusing on studying such communication network characteristics as delay, range, throughput, and network capacity is available [21]. The authors of [22] propose using fuzzy logic to optimize the adaptive data rate (ADR) for energy-efficient LoRaWAN communication. Additionally, emerging studies explore the possibility of reducing the energy consumption of edge devices operating in low-energy LoRaWANs by optimizing the deployment of gateways in a defined area. The authors of [23] investigated the most suitable locations for gateways, while considering limited radio resources and energy consumption. A model was proposed for minimizing energy usage by periodically scheduling changes in gateway locations to extend network lifetime. Researchers are also exploring the feasibility of a scalable probabilistic approach that can achieve service access control [24] and efficient sharing of the LoRaWAN between different services. The proposed approach demonstrates the ability to efficiently utilize the limited resources of a radio network, while minimizing the energy consumption of the network elements. The authors of [25] address the development of new methods for energy saving in IoT devices used in medicine. Such devices are used to collect data, monitor health, and diagnose patients. The authors proposed a novel, avant-garde framework to simultaneously optimize energy efficiency, battery life, and reliability for smart and connected healthcare. An adaptive transmission data rate (ATDR) mechanism was proposed, which controls the average constant energy consumption by varying the active time of the sensor node to optimize energy over the dynamic wireless channel. A self-adaptive routing algorithm (SARA) has also been proposed. The authors have proven that their solutions outperform standard solutions in terms of better reliability and battery life. The authors of [26] presented an improved energy consumption model for a LoRaWAN node based on in situ measurements. This improved model takes into account the number of nodes in the network, the probability of collisions, which depends on sensor density, and the number of retransmissions. The results show the effect of the number of nodes in a LoRaWAN on a node's energy consumption, demonstrating that the number of sensors that can be integrated into a LoRaWAN is limited due to the probability of collisions.

The analyzed scientific articles show various studies aiming to demonstrate the possibility of reducing energy consumption. Most of them focus on modifying the protocol or organization of IoT networks. In this article, the authors present an alternative approach to reduce energy consumption by predicting measurement values, limiting data transmission, and modulating antenna power. The objective of the study is to prove that, without interfering with the LoRaWAN protocol, further reductions in the energy consumption of end-node devices operating in a low-energy network can be achieved.

In summary, LoRaWAN is an emerging technology that, with its low power consumption and long range, appears to be an ideal candidate for use in IoT devices. In addition, opportunities in LoRaWAN to reduce the energy consumption of edge devices are demonstrated, which can implement certain algorithms to optimize energy consumption, despite their computing power limitations.

3. LoRa Wireless Communication Technology and LoRaWAN Wireless Communication Protocol for IoT Applications

LoRa is a long-range, low-power wireless communication system that can transmit small amounts of data over long distances. These two features make it an attractive solution for use in IoT industries. A LoRaWAN is a protocol developed based on LoRa modulation. It is a software layer that determines how devices use LoRa hardware, that is, when they transmit data and in what format. Their long communication range and low power consumption also make LoRa an ideal solution for industrial IoT applications. In addition, LoRa overcomes many new challenges in rural and urban areas, such as those related to climate, pollution, or natural disasters, as well as the metering of energy consumption or production in renewable energy systems [27,28]. Due to its simplicity and cost effectiveness, this solution is widely used in transportation, manufacturing, agriculture, household appliances, and in some cases, even in wearables. LoRaWANs have also been studied for their potential use in communication with various satellite systems, as well as high-altitude platform stations (HAPS) in the stratosphere. The authors of [29] demonstrate the usefulness of this type of communication through a suitable ISM (industrial, scientific, medical) band propagation model. LoRa uses an electromagnetic wave modulation technology referred to as chirp spread spectrum (CSS) [30]. This technique has been used for decades in military and aerospace applications. The technique uses the encoding of information on radio waves using chirp pulses. This modulation is extremely resistant to interference and can be received over long distances depending on hardware capabilities. LoRa is ideal for applications that transmit small amounts of data at low speeds. Data can be transmitted over longer distances than with other wireless technologies, such as Wi-Fi, Bluetooth, or ZigBee. This makes LoRa ideal for transmitting sensor data in low-power modes. LoRa uses unlicensed sub-gigahertz bands, such as 915 MHz, 868 MHz, and 433 MHz. It can also operate at 2.4 GHz for much higher data rates, albeit with a more limited range. These frequencies belong to the ISM bands, which are globally reserved for industrial, scientific, and medical purposes. LoRaWANs are the software layers that determine how devices use LoRa hardware, such as when and in what format they transmit data. The protocol itself is developed and maintained by the LoRa Alliance. The LoRaWAN specification is a standard that allows the seamless integration with third-party devices. This single factor is one of the reasons why LoRa technology has rapidly developed in the IoT industry. LoRa and LoRaWAN can connect the wide and dense networks of edge devices, making it possible to capture and monitor data from thousands of nodes in a manageable way.

The LoRaWAN architecture consists of four main components:

- End-points (end-node);
- Gateway;
- Network server;
- Application server.

End-points are devices located at the end of a LoRaWAN. They are most often equipped with sensors for periodic data collection and monitoring. They often consist of a

low-power microcontroller, which has been used in this field for several years. They do not require maintenance and are equipped with low-power LoRa transmitters to send data packets to gateways. End devices can be classified into three different types as follows:

- Class A;
- Class B;
- Class C.

Class A: This device creates two message reception windows after each message is sent. The first window opens after the node finishes transmitting the message at time T_1 , while the second window opens after the T_1 window closes at time T_2 (see Figure 1). The T_1 and T_2 times depend on the region in which the device is used and are described in the LoRaWAN standard documentation. Crucially, aside from the two windows after sending a message to the network, there is no other way to transmit messages from the network: Tx is sending data, and Rx is receiving data.

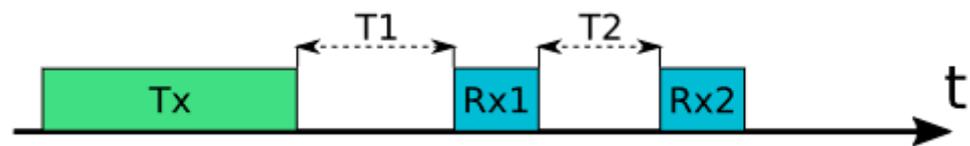


Figure 1. Transmission waveform for a Class A node (own source).

Class B: This device functions in a similar way to Class A, except T_b message reception windows are opened at specific time intervals (see Figure 2). The time is synchronized by signals, i.e., beacons sent by the gateway. This solution allows for a two-way communication between the node and the network with a limited power consumption.

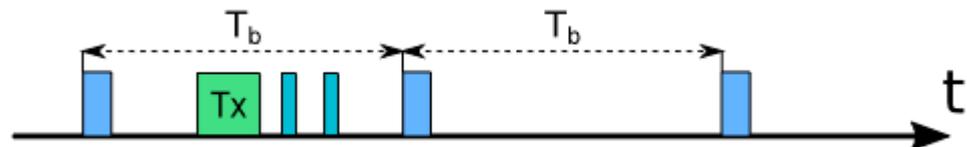


Figure 2. Transmission waveform for Class B node (own source).

Class C: This device constantly waits for incoming messages, the exception being data transmission from the node to the network (see Figure 3). Class C devices require a relatively significant amount of power since the radio module is constantly operating in receive or transmit mode.



Figure 3. Transmission waveform for a Class C node (own source).

A LoRaWAN gateway is a bridge between nodes and the network. They receive information from end-points via a LoRa hub and then transmit these data to a network server via the Internet or private network infrastructure. Their key function is to forward packets. Figure 4 shows a LoRa gateway, the LORANK-8, manufactured by Ideetron and used for the research. It uses an 868 MHz frequency for radio transmission. The gateway allows the simultaneous processing of up to eight signals and can connect to approximately 60,000 end devices. The receiver sensitivity is -138 dBm, while antenna amplification is possible at 500 mW. The values described by the manufacturer provide a range of up to 25 km in an open space.



Figure 4. View of LoRa gateway: LORANK-8 (own source).

The gateway used for this study (Figure 4) was constructed using a LoRaWAN iC880A-SPI hub (IDEETRON B.V., Doorn, The Netherlands) [31] and a single-board BeagleBone low-voltage computer [32] with open-source software. The iC880A can receive up to eight LoRa protocol frames sent simultaneously on different channels using different spreading factors. This allows easy-to-use star or multi-star networks to be constructed without the need for routers or repeaters used in short-range networks. The hub has two Semtech SX1257 radios, used as transceiver units in the device. Signal analysis and processing on multiple channels simultaneously were carried out using the SX1301 processor.

The web server consolidates all data received from the gateways and sends these data to the application server.

The application server allows for the visual or analytical interpretation of the data. Data points can also be integrated into the platform to take various actions, such as controlling actuators. It is even possible to set up a notification service to alert an engineer about a potential problem.

One of the advantages of a LoRaWAN is the ability to position it without the need for GPS receivers. This is particularly useful when tracking assets or even sensors, as it consumes much less battery power and is cheaper to implement than traditional methods. A LoRaWAN is designed for large IoT networks, where thousands of devices can be connected, even to a small number of gateways. These gateways can listen on multiple channels and process numerous messages simultaneously. A key feature of LoRaWAN is the speed at which data can be sent. A provision is made for different transmission speeds that can be used to transmit data. As a result, slower transmission allows for a more reliable transmission of information over longer distances. A key feature of LoRaWAN is that the network can automatically optimize the speed at which a device transmits data. This feature is called the adaptive data rate (ADR) and is key to increasing the aggregate throughput of a LoRaWAN. ADR can easily expand a network by increasing its number of gateways. This will cause many devices to automatically change their SF, and thus reduce the time it takes to transmit a packet, in effect leaving more free “ether” for other devices. ADR is a very simple mechanism that leads to a change in transmission rate based on the following principles:

- If the power of the radio signal (called the link budget) is high, the transmission rate can be increased;
- If the link budget is small, the transmission rate can be decreased.

For security, a LoRaWAN uses two layers of security. One at the network level and one at the application level. Network-level security allows you to be sure that a given end

device is the intended device, while application-level security ensures that the network operator cannot access user data. A LoRaWAN uses AES encryption and key exchange. The network layer is responsible for identifying nodes, checking that messages are coming from specific devices, and is referred to as network integrity checking. It also implements MAC command encryption. At the application layer, the encryption and decryption of the actual useful data are possible. Both keys are encrypted with the 128-bit AES algorithm in Electronic Codebook (ECB) mode.

In 2021, the LoRa Alliance, a global association of companies supporting the LoRaWAN standard for IoT announced that the LoRaWAN had been officially approved as a communications standard for LPWAN by the International Telecommunication Union (ITU) [33].

4. Proposal of a Test Set for Measuring the Energy Consumption of End Devices Operating in a LoRaWAN and Optimization Algorithms

The Raspberry Pi Pico platform [34] was used to construct the final device. The platform is equipped with a dual-core ARM Cortex M0+ processor, which reaches up to 133 MHz. The device has 2 MB of flash memory and 264 kB of SRAM. The RFM95W module [35] was chosen to implement radio communication. The radio module supports the LoRa protocol and has a high sensitivity of -148 dBm, which increases the range of the end device. The SHT40 [36] was used as the measuring sensor. It provides a temperature measurement between -40 °C and 125 °C and a humidity measurement with an accuracy of 1.8% RH. The measurement data are transmitted via a radio module to a LoRa gateway, and then transmitted to a cloud server via a router and the Internet. The proposed architecture of the low-energy communication system with a LoRaWAN, along with the measurement apparatus, is shown in Figure 5.

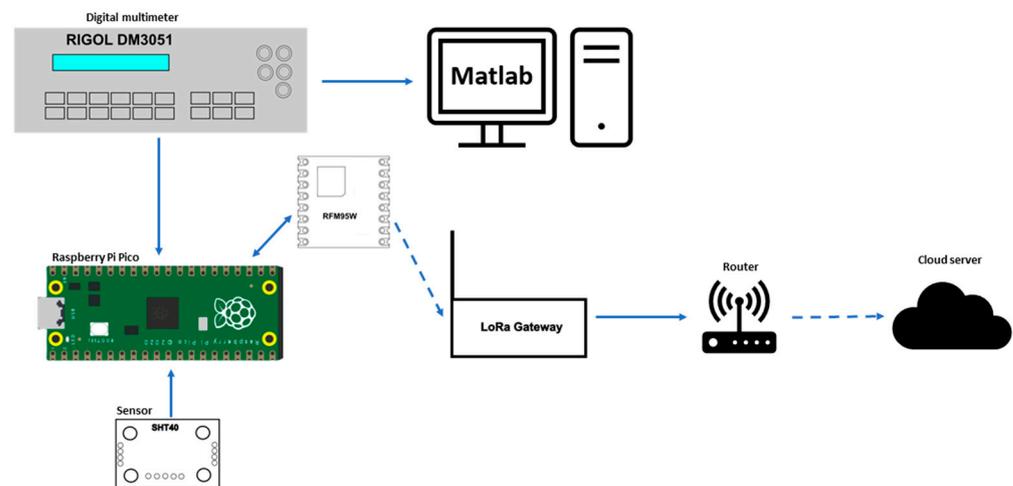


Figure 5. The architecture of the communication system and measuring equipment (own source).

4.1. The Proposal of Temperature Value Prediction Algorithm

In the energy consumption study, it was assumed that the temperature measurement was implemented in each iteration of the main loop of the program. In the first ten iterations of the main loop of the program, an object of JSON type was created, containing information on the current temperature measurement and the sleep time of the device between successive measurements. Sending information regarding the set delay allows the server to control the correctness of the received data and overwrite the results if the predicted measurement is greater than the specified measurement error. Putting the device to sleep is intended to reduce the energy consumed by the microcontroller between sending data to the server. Once the measurement buffer was full, the temperature value was predicted. The predicted measurement was attached to a JSON structure and then sent to the server. The RapidJSON library [37] was used to prepare the JSON object, which

provides an object-oriented method of filling the structure with data. Due to limitations in the size of the transmitted data dependent on the spreading factor, the measurement results were rounded to two decimal places using the `SetMaxDecimalPlaces` function [37]. The message with the predicted value was sent to the server by the end device, which should contain the current reading, predicted reading, and sleep time of the microcontroller. The data were sent at every second iteration of the main program loop if no deviation from the predicted measurement was recorded. When there was a difference between the measurements, the data were sent in the current iteration. The structure of the JSON file was read on the server, and the values of the measurements were saved as correcting values of the previously received packet or as the current and predicted value. For this purpose, the time taken to save the received data was verified. If it is less than twice the value of the delay parameter from the JSON structure, the previous value was updated, and the current measurement was saved. This makes it possible to eliminate errors in the history of received data from the terminal device.

The purpose of implementing the module responsible for predicting measurement values was to optimize the energy consumption of the end device. During testing, it was noticed that the highest energy consumption occurs during the sending of packets via the radio transmitter. One way to accumulate energy could be to reduce the number of packets sent by attaching the predicted value to the current value.

A value must be predicted based on a predetermined model. In the project, the data, which are set for training the model, were taken from the measurements recorded by the temperature sensor. The prediction of the temperature value was implemented through a linear prediction model. This is one of the simplest algorithms to implement, and the application of the studied value is sufficient. Figure 6 shows a plot of the linear equation, whose parameters a and b are determined based on previous measurements.

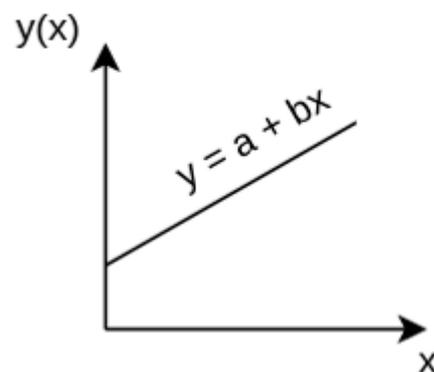


Figure 6. Linear equation for prediction (own source).

The first step of the prediction is to collect data on previous measurements. A collection containing ten previous values proved to be sufficient to correctly determine subsequent values. In the first ten iterations of the program, measurements are added one at a time to the `temp_measurements` buffer. If the array is filled according to its declared size, the parameters necessary for prediction are determined. In the condition shown in Listing 1, the absolute value of the temperature is also checked. If the predicted temperature differs from the measured temperature by the `MEASUREMENT_CORRECTNESS` parameter set to 2%, the model is redetermined. In the first iteration, this condition is always true due to the configuration of the initial values beyond the sensor's measurement accuracy.

The function `update_params_for_prediction`, shown in Listing 2, is responsible for training the model, i.e., determining the values of a and b necessary to solve the linear equation. The parameters input to the function are a buffer containing previous measurements, and parameters a and b are defined as a `std::pair` type. The process of determining the values involves minimizing the error calculated at each epoch, and 200 epochs were sufficient to determine the parameters. In the function code, however, the number of loops is 2000: the

product of the number of epochs and the measurements in the buffer. An alpha factor is also used to determine the parameters, controlling the rate at which the parameters are adjusted to near-optimal values. It is also referred to as the learning rate (learning rate). The result of the loop is the determined values of the parameters a and b.

Listing 1. Implementation of the condition required to create a model.

```
if (
    temp_measurements.size() > SIZE_OF_MEASUREMENTS_BUFFER &&
    abs(temp.temperature - previous_predicted_temp) >
    (temp.temperature * MEASUREMENT_CORRECTNESS)
){
    update_params_for_prediction(temp_measurements, params);
    iter_for_prediction = 0;
}
```

Listing 2. Value update function of the parameters a and b in the linear equation.

```
static void update_params_for_prediction(
    std::vector<double>& buffer,
    std::pair<double, double>& params
){
    /*Intialization Phase*/
    double err;
    double alpha = 0.01;
    params.first = 0.00;
    params.second = 0.00;

    /*Training Phase*/
    for (int epoch = 0; epoch<2000; epoch++)
    {
        int idx = epoch % SIZE_OF_MEASUREMENTS_BUFEER;
        double p = params.first + params.second * idx;
        err = p - buffer[idx];
        params.first = params.first - alpha * err;
        params.second = params.second - alpha * err * idx;
    }

    iter_for_prediction = 0;
}
```

Parameter redetermination only occurs when the acceptable error threshold is exceeded. Until that point, the program considers the determined parameters to be correct and makes a prediction of subsequent values. The function that makes this possible is *predict_temp_value*. Based on the linear equation formula and parameters a and b, the predicted temperature is determined, as shown in Listing 3.

Listing 3. Predictive function of temperature values.

```
double predict_temp_value(float& actual_temp, std::pair<double,
    double>& params)
{
    double predicted_value = params.first +
        (params.second * (SIZE_OF_MEASUREMENTS_BUFFER +
            iter_for_prediction));
    return predicted_value;
}
```

The form of the complete algorithm for predicting temperature values is shown in Algorithm 1.

Algorithm 1 Temperature prediction algorithm

```
# statement of constants
size_of_measurements_buffer ← 10
measure_correctness = 0.02
temp_measurements ← [ ]
predicted_temp ← −150
previous_predicted_temp ← −150
alpha ← 0.01
a ← 0.00
b ← 0.00
iter ← 0

# main loop declarationwhile is_main_loop()
temp ← get_from_sensor()

# checking the condition (exceeding the error threshold)
if length(temp_measurements) > size_of_measurements_buffer
&&absolute(temp-previous_predicted_temp) > (temp × measure_correctness)
then
    iter ← 0
    a ← 0.00
    b ← 0.00

# determination of the number of epochs
for epoch ← 0 to amount_of_epoch
    idx ← epoch % size_of_measurements_buffer
    # determination of prediction error
    error ← a + b × idx
    a ← a − alpha × error
    b ← b − alpha × error × idx
if a ≠ 0 && b ≠ 0
then
    previous_predicted_temp = predicted_temp
    predicted_temp ← a + b × (size_of_measurements_buffer × iter)
    temp_measurements.erase(temp.measurement.begin())

iter ← iter + 1
temp_measurements[length(temp_measurement)] ← temp
```

4.2. The Proposal of Optimization of Antenna Gain Values Algorithm

The power of the transmitting signal is a factor that directly affects the power consumption of the end device. Up to the moment the end device joins the LoRa network, the antenna gain is maximized; however, after that it is optimized. Each packet received from the server has a value referred to as Received Signal Strength Indication (RSSI), an indicator of the received signal strength, which can be used as a value to determine the required antenna gain. The closer the RSSI value is to zero, the stronger the signal strength increases.

The *adjust_antenna_power* function, shown in Listing 4, is called after receiving each successive packet sent from the server. The action of the method is to reduce the gain of the antenna if the value of the signal factor is outside the designated limits. If the signal is at the limit of obtaining a connection, the value of the *tx_power* parameter is reduced by 1, which causes an increase in the antenna gain. The opposite action is carried out when the RSSI is below −100 dBm. The functions *increase_tx_power* and *decrease_tx_power* can reduce or increase the value of the parameter; however, this value can only vary between *TX_POWER_0* and *TX_POWER_6*.

Listing 4. Functions defining the value of antenna gain depending on RSSI.

```

int set_antenna_power(int rssi)
{
    if (rssi <= -100) return TX_POWER_0;
    else if (rssi <= -85 && rssi > -100) return TX_POWER_1;
    else if (rssi <= -70 && rssi > -85) return TX_POWER_2;
    else if (rssi <= -55 && rssi > -70) return TX_POWER_3;
    else if (rssi <= -40 && rssi > -55) return TX_POWER_4;
    else if (rssi <= -20 && rssi > -40) return TX_POWER_5;
    else if (rssi > -20) return TX_POWER_6;

}

int increase_tx_power(int tx_power)
{
    if(tx_power > TX_POWER_0) return tx_power--;
    return tx_power;
}

int decrease_tx_power(int tx_power)
{
    if(tx_power > TX_POWER_6) return tx_power++;
    return tx_power;
}

int adjust_antenna_power(int rssi, int tx_power)
{
    if (rssi < -130) return increase_tx_power(tx_power);
    else if (rssi > -100) return decrease_tx_power(tx_power);

    return tx_power;
}

```

Listing 5 shows the conditions in the main loop of the program responsible for sending and receiving packets. The results of a valid message sent by the *lorawan_send_unconfirmed* function are a value greater than zero. The *ANTENNA_ITER_THRESHOLD* parameter defines the maximum number of sent packets after which the end device did not receive a response from the server. The comparison of this parameter, along with the variable incremented after each message is sent, allows you to control whether the communication between the end device and the server is two-way. The application server should, in response to every tenth packet received, send a feedback message containing RSSI. Unfortunately, there is a risk that the message, due to interference or lack of coverage, will not reach the recipient. It is therefore necessary to check the end device for the several packets that have been sent from the server. For this purpose, the parameter *ANTENNA_ITER_THRESHOLD* was used, which is set to 20. This means that if no response is received from the server for twenty messages sent, then the antenna gain should be increased. The incremented variable is *iter_for_antenna_gain*, which takes the value 0 after receiving a packet from the server.

Listing 5. Conditions found in the main loop of the program.

```

if (lorawan_send_unconfirmed(message.GetString(),
    strlen(message.GetString()), 2, antenna_gain
) < 0)
{
    printf("failed!!!\n");
}
else
{
    if(iter_for_antenna_gain == ANTENNA_ITER_THRESHOLD)
    {
        antenna_gain = TX_POWER_0;
        iter_for_antenna_gain = 0;
    }else {
        iter_for_antenna_gain++;
    }
    gpio_put(PICO_DEFAULT_LED_PIN, true);
    printf("success!\n");
}

if (lorawan_process_timeout_ms(RECEIVE_DELAY) == 0)
{
    //check if a downlink message was received
    receive_length = lorawan_receive(
        receive_buffer, sizeof(receive_buffer), &receive_port
);
    if (receive_length > -1 && lorawan_rx_rssi() != NULL)
    {
        iter_for_antenna_gain = 0;

        if (!antenna_gain_setted) {
            antenna_gain = set_antenna_power(lorawan_rx_rssi());
            antenna_gain_setted = true;
        } else {
            antenna_gain = adjust_antenna_power(lorawan_rx_rssi(),
                antenna_gain);
        }
    }
}

```

The form of the complete algorithm for optimizing antenna gain values is shown in Algorithm 2.

Algorithm 2 Antenna gain optimization algorithm

```

# statement of constants
antenna_iter_threshold ← 20
iter_for_antenna_gain ← 0

# main loop declaration
while is_main_loop()
    iter_for_antenna_gain ← iter_for_antenna_gain + 1

    if iter_for_antenna_gain == antenna_iter_threshold
        then antenna_gain ← 0

# procedure for determining optimal antenna gain
if is_response_from_gateway()
    then
        rssi ← get_rssi_from_gateway_response()
        if antenna_gain_setted
            then
                if rssi ≤ −100
                    then antenna_gain ← 0
                    else if rssi ≤ −85 && rssi > −100
                        then antenna_gain ← 1
                else if rssi ≤ −70 && rssi > −85
                    then antenna_gain ← 2
                else if rssi ≤ −55 && rssi > −70
                    then antenna_gain ← 3
                else if rssi ≤ −40 && rssi > −55
                    then antenna_gain ← 4
                else if rssi ≤ −20 && rssi > −40
                    then antenna_gain ← 5
                else
                    then antenna_gain ← 6
            else
                then
                    if rssi < −130
                        then
                            if antenna_gain > 0
                                then antenna_gain ← antenna_gain-1
                            else if rssi > −100
                                then
                                    if antenna_gain < 6
                                        then antenna_gain ←
antenna_gain + 1

```

5. Testing of the Developed Algorithms to Optimize Antenna Gain and Predict Temperature Value

The developed algorithms were subjected to testing due to the energy consumption of the terminal equipment. The tests were carried out under laboratory conditions. The measuring apparatus (RIGOL DM3051 digital multimeter [38]) had two measurement channels, which allowed changes in the voltage of the power supply system and current on the microcontroller board to be simultaneously monitored. A YwRobot MB-V2 module [39] was used to power the circuit, with an output voltage of 5 V and a maximum output current of 700 mA. The circuit along with the measurement apparatus is shown in Figure 7. The tests were carried out by recording the samples for 600 s after power-up, allowing 30,000 current readings to be recorded. The collected results were saved on a portable device in csv format. Before each measurement was taken, the server settings were reset to ensure that conditions were the same, and thus reliable results were obtained. Each

of the performed measurements included fluctuations in the changes studied during the acceptance of the end device by the server and during the sending of data to the server. The distance between the network gateway and the end device was four meters, and there were no obstacles that could interfere with the signal.

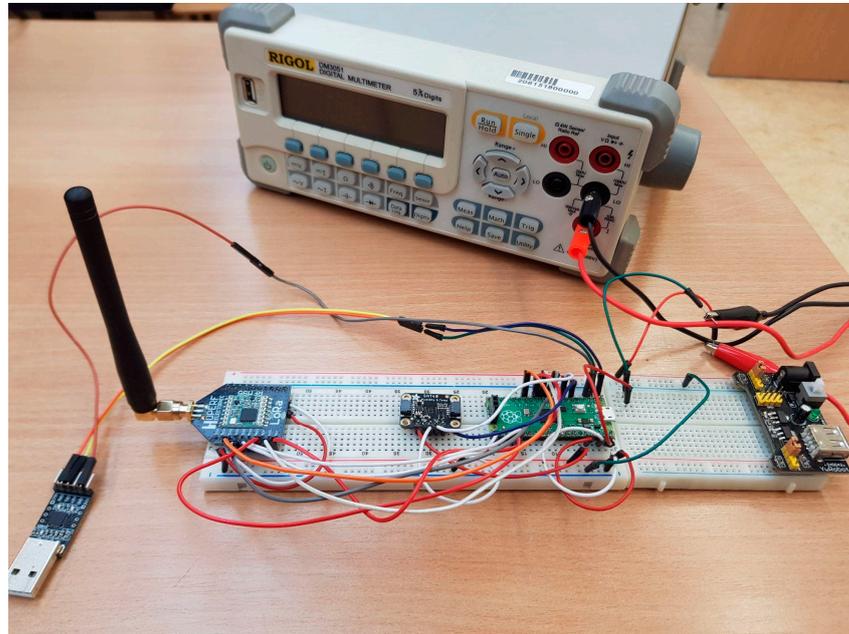


Figure 7. Laboratory set with measuring instruments (own source).

In the first step, the test scenario involved investigating changes in the electrical current consumption of the terminal device, using only a digital multimeter, for the following cases: without optimization methods applied, with temperature measurement prediction applied, with antenna gain optimization applied, and with temperature measurement prediction and antenna gain optimization applied. The measurement values were saved on the SD card. The test scenario was repeated several times, and reproducible results were obtained. A decision was made to perform one series of tests according to the scenario, and thus the results obtained were analyzed.

During the study, it was assumed that 868 MHz (free ISM band) would be used for communication. The built end node used only a standard CSS signal modulation. When testing the effectiveness of the measurement prediction algorithm, the power levels did not change and the distance between the end-node and the gateway was not altered. According to the principles of the LoRaWAN protocol, it was assumed that the data rate would be automatically optimized. According to the data published in the protocol, it was assumed that the transfer would range from 37.5 kb/s to 11.5 b/s.

Figure 8 shows the current measurement results for the case without optimization mechanisms: the standard operating mode of the edge device.

Figure 9 shows the current measurement results for the case with the implemented temperature value prediction algorithm. The peaks, numbered from 3 to 7 and visible in the graph, show the moment of data accumulation by the buffer with predictive values, which is the basis for calculating the predicted temperature. When comparing the results with the values in Figure 8, it can be seen that the frequency of sent data halves confirms the operation of the prediction algorithm. One measurement was sent again as a result of an excessive deviation between the measured and predicted temperature. The difference was recorded, and a packet was sent by the end device, which can be observed in peak no. 25 on the graph in Figure 9.

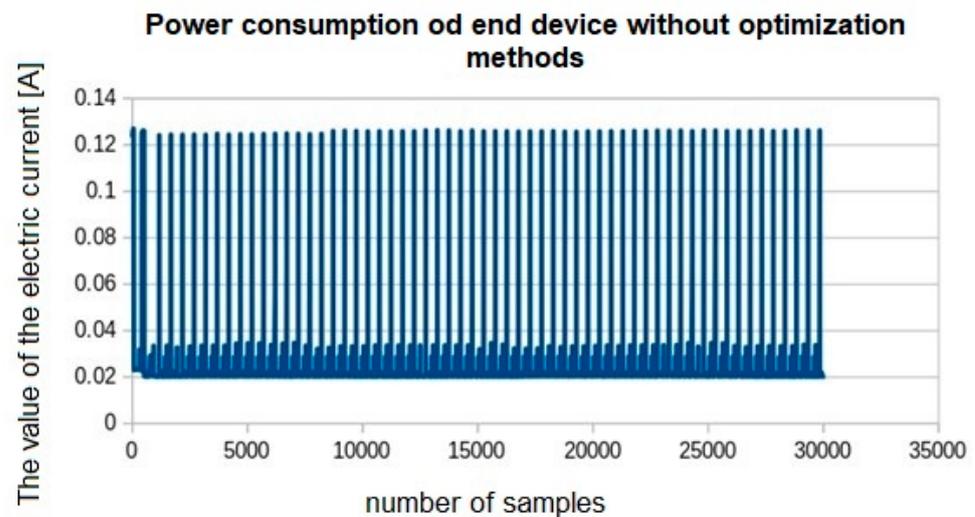


Figure 8. Graph of electricity consumption of the end device without the use of optimization methods.

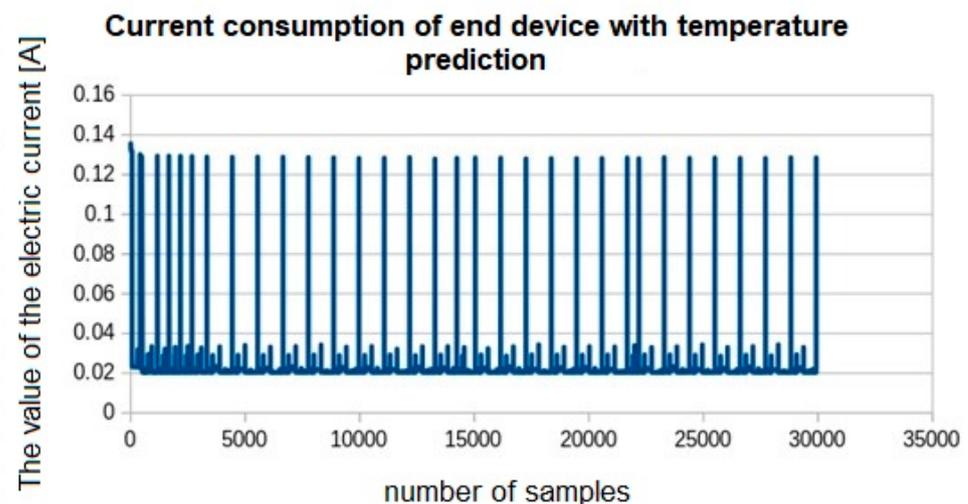


Figure 9. Graph of electricity consumption of the end device using temperature prediction.

Another test was conducted using antenna gain optimization. The results shown in the graph in Figure 10 indicate a reduction in power consumption from the third data transmission through the device. The graph in Figure 10 shows three power consumption thresholds, which were formed based on the values of local maxima. The first of these relates to the device's authorization in the LoRaWAN. The second threshold was established based on the initial optimization of the antenna gain. The third threshold was obtained by making a gain adjustment based on the received RSSI value from the application server.

The last measurement was carried out with a temperature prediction and antenna gain optimization implemented. The result was a halving of the frequency at which the radio network sends data and reduces current consumption due to the reduction in gain. The test results are shown in Figure 11.

Each of the measurements made with the optimization implemented showed that it was possible to reduce the power consumption of the tested laboratory set. The reduction in power consumption only refers to the moment when a data frame was sent over the radio network; however, the differences are noticeable. A comparison of the average power consumption of the end device during the measurements is illustrated in Figure 12. The results illustrate the average power consumption expressed in watts (W) during a measurement lasting 600 s.

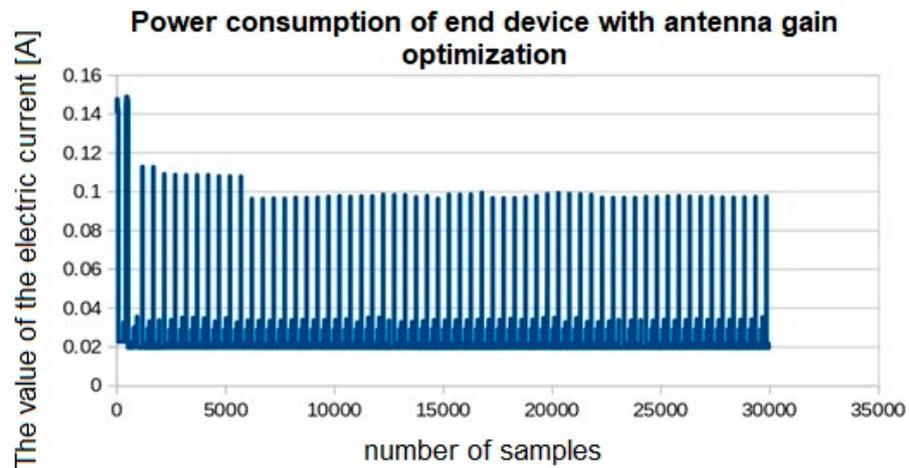


Figure 10. Graph of electricity consumption of the end device using antenna gain optimization.

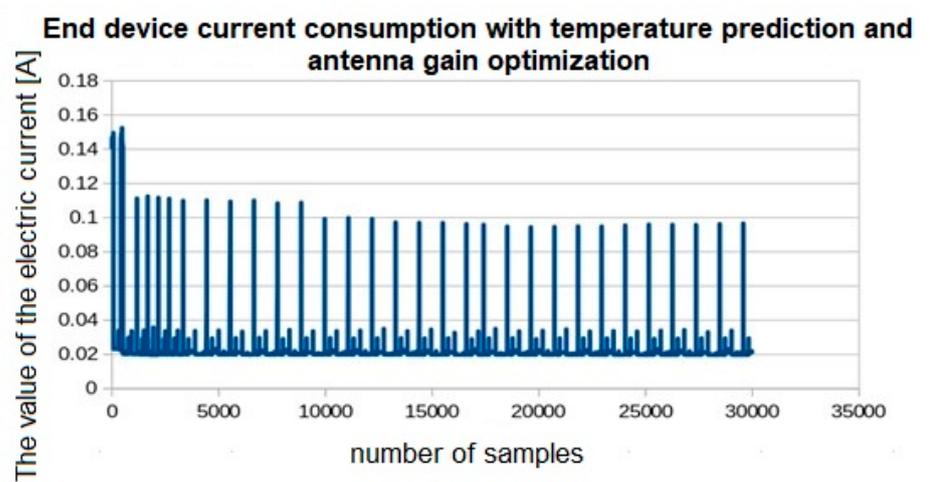


Figure 11. Graph of electricity consumption of the end device using antenna gain optimization and temperature prediction.

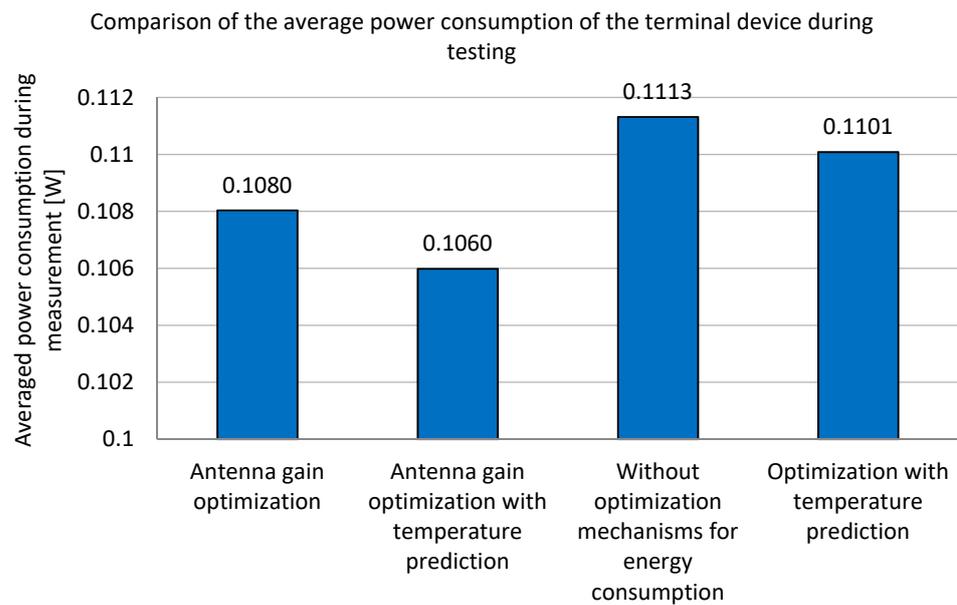


Figure 12. Comparison of the average power consumption of the terminal device for each measurement.

The percentage reduction in energy consumption of the end device with the implemented algorithms, compared to communications without the implemented optimization methods, is shown in Figure 13. The temperature prediction appeared to reduce energy consumption by only 1.12%, despite the noticeable differences in the number of peaks shown in the graph, illustrated in Figure 9. The optimization of the antenna gain results in a greater reduction in the energy consumption of the end device: 3.04%. An experiment using both algorithms yielded the greatest reduction in energy consumption: 5.03%. In related studies [15–19], the authors achieved a reduction in energy consumption by means of various modifications to the communication protocol or by proposing their own optimization algorithms. The reduction in power consumption ranged from 5 to 15% compared to standard LoRaWAN communication.

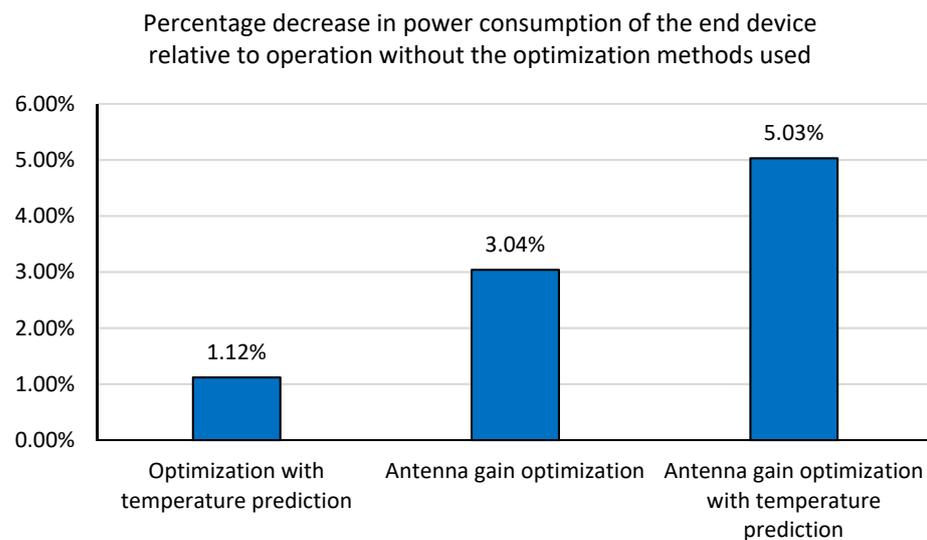


Figure 13. Graph showing the percentage decrease in power consumption of the end device as a result of the applied optimization algorithms.

In a second series of tests of the proposed algorithms to optimize energy consumption, the measurement apparatus (RIGOL DM3051 digital multimeter) was connected to the Matlab platform (Poznan University of Technology academic use) [40]. The Matlab Instrument Control Toolbox was used to communicate between the measurement apparatus and the Matlab platform, and an Interchangeable Virtual Instrument (IVI) [41] for the measurement apparatus was imported. A script (M-file) was prepared to record the measurements. The measurement data were saved to files (MAT-files) in the Matlab platform for further analysis.

Studies of the reduction of energy consumption in each of the proposed algorithms depending on changes in the distance between the end device and the LoRa gateway inside the building, the Lecture and Conference Centre of the Poznan University of Technology, have been done. In selected locations, the number of building partitions was changing. A permanent location for the end device with measuring equipment and a computer with the Matlab platform was proposed. During the tests, the position of the LoRa gateway was changed. The end device position and LoRa gateway positions are shown in Figure 14. Point 1 is the end device position. Points 2, 3, and 4 are the gate positions.

General principles and configuration of test scenarios:

1. (A): Distance between points 1 and 2 = 8 m, two walls;
2. (B): Distance between points 1 and 3 = 20 m, three walls;
3. (C): Distance between points 1 and 4 = 24 m, four walls;
4. (I): Measuring electric current consumption and energy consumption for the case without optimization methods applied;

5. (II): Measuring electric current consumption and energy consumption for the case with temperature measurement prediction applied;
6. (III): Measuring electric current consumption and energy consumption for the case with antenna gain optimization applied;
7. (IV): Measuring electric current consumption and energy consumption for the case with temperature measurement prediction and antenna gain optimization applied;
8. Measurement recording time in the Matlab platform—100 s, 50 Hz sampling.

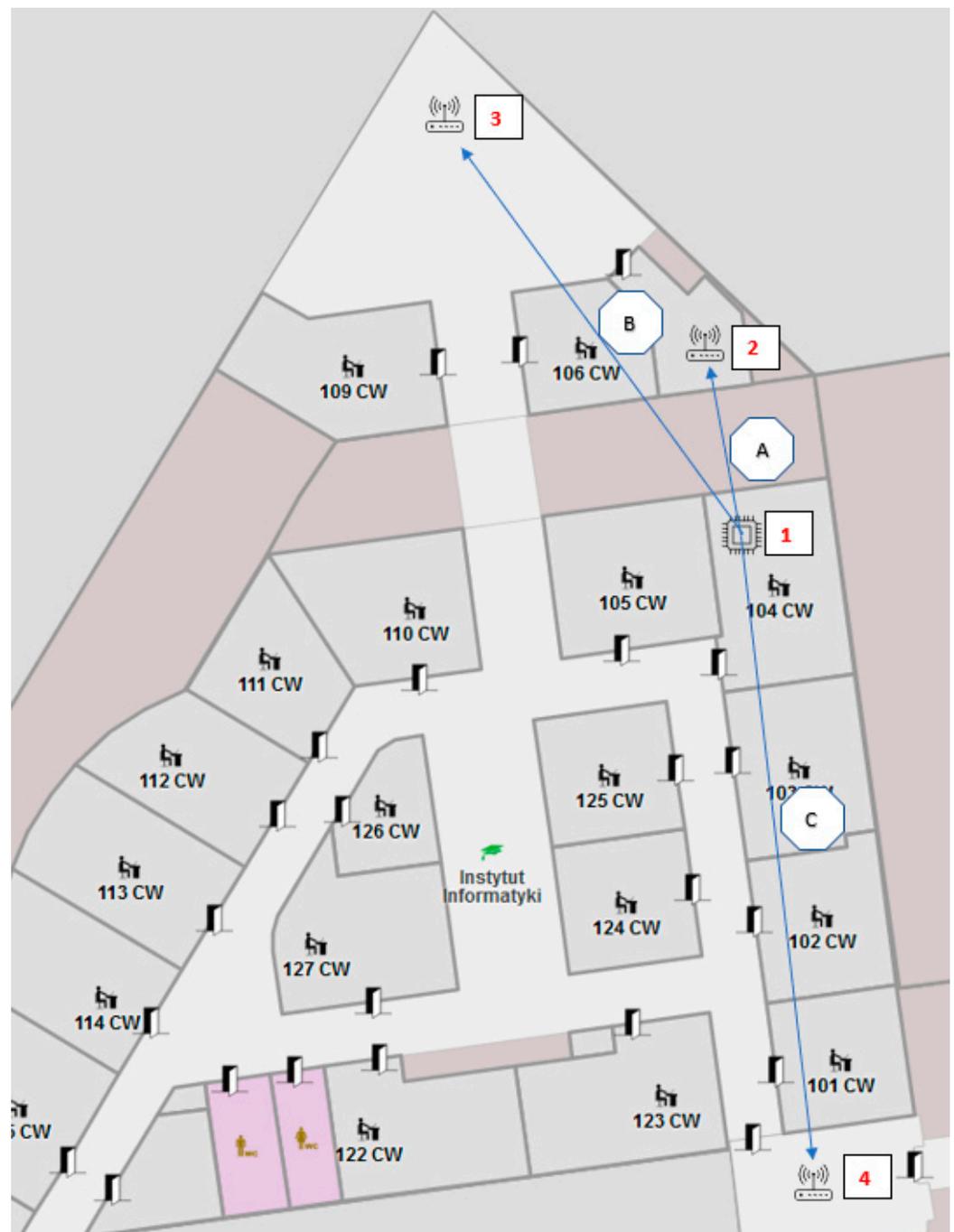


Figure 14. End device position (1), LoRa gateway positions (2, 3, 4) and distances between end device and LoRa gateway (A, B, C) in Lecture and Conference Centre of the Poznan University of Technology—first floor.

The energy consumption of the optimization algorithms was studied. The effect of distance and number of walls on RSSI values and the number of data packets not lost was also investigated. Metadata attached to measurements sent to The Things Stack open web server was used [42]. The RSSI and number of packets-received tests involved the end device sending temperature measurements from each of three locations in the building. The maximum and minimum RSSI values were read, and the median was calculated through the built-in Matlab Data Statistics tool. The percentage of data packets not lost was also calculated. Communication study results for the following cases: without optimization methods applied—designation in the table and in the graphs (I), with temperature measurement prediction applied—designation in the table and in the graphs (II), with antenna gain optimization applied—designation in the table and in the graphs (III), and with temperature measurement prediction and antenna gain optimization applied—designation in the table and in the graphs (IV), for three different device configurations in the building (A, B, C) are shown in Table 1.

Table 1. RSSI and received packets test results for the four cases.

Configuration in Building	RSSI _{MIN} [dBm]				RSSI _{MAX} [dBm]				Median [dBm]				Received Packets [%]			
	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
1–2 (A)	−55	−56	−56	−55	−47	−47	−48	−47	−51	−51	−52	−51	99	98	99	99
1–3 (B)	−69	−69	−69	−69	−58	−58	−58	−59	−64	−64	−64	−64	100	100	100	100
1–4 (C)	−77	−78	−77	−77	−66	−66	−66	−67	−71	−71	−71	−71	99	98	99	99

The analysis of the values collected in Table 1 shows that the implementation of the developed algorithms for the optimization of energy consumption does not affect the RSSI values and the number of lost data packets for different configurations of equipment in the building, which is a positive result of the test of the algorithms. At the same time, the electric current consumption of the end device was analyzed during each of the test scenarios for the power consumption optimization algorithms. The electric current consumption data were transferred to the Matlab platform and were analyzed. The electric current consumption of the end device for the four cases (I, II, III, IV) and for the three locations of the end device and LoRa gateway (A, B, C) are shown in Figures 15–17.

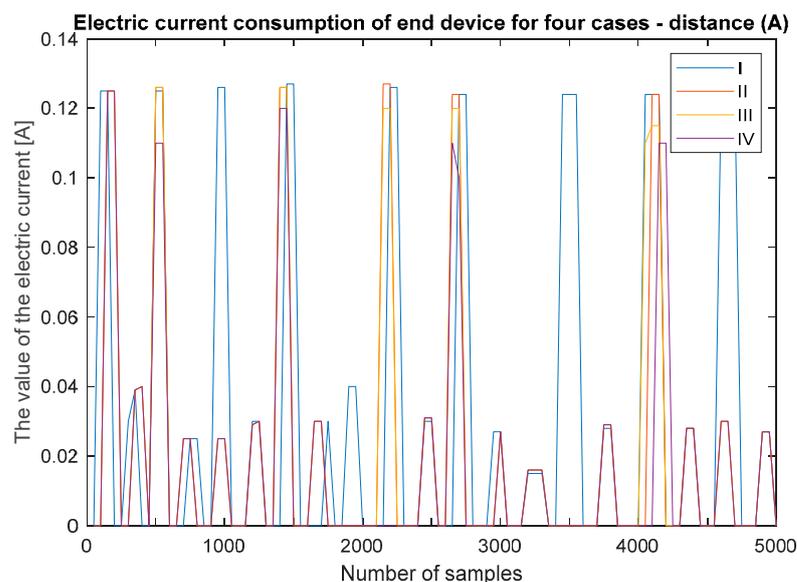


Figure 15. Graph of electric current consumption of the end device for four cases and devices—locations (A).

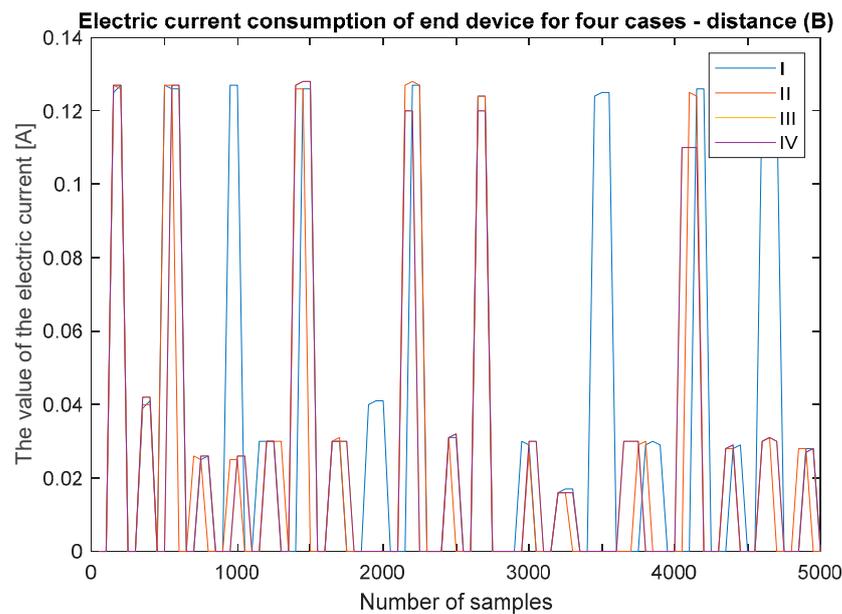


Figure 16. Graph of electric current consumption of the end device for four cases and devices—locations (B).

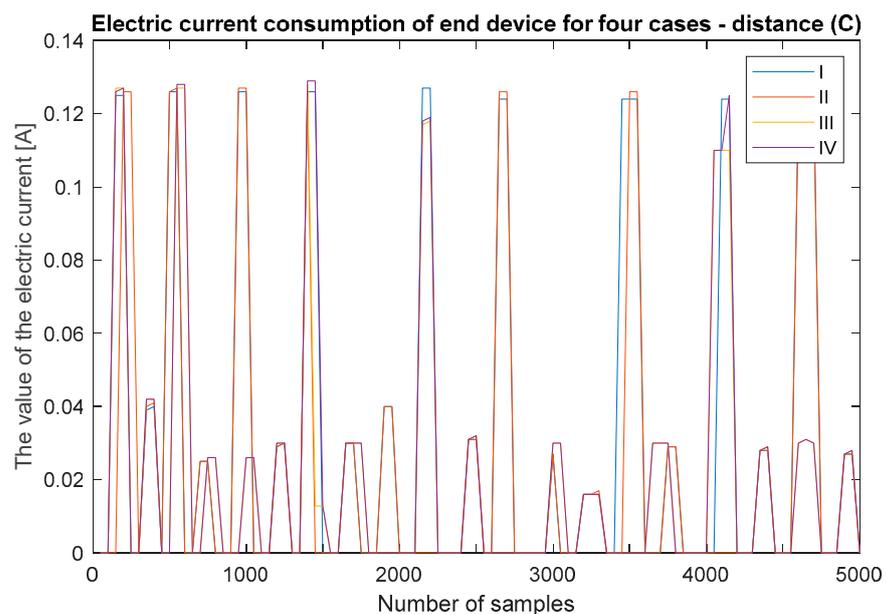


Figure 17. Graph of electric current consumption of the end device for four cases and devices—locations (C).

In Figures 15–17, the highest electric current consumption is seen for the case without optimization algorithms—case (I). A reduction in electric current consumption by lowering its value and the frequency of communication resulting from the optimization algorithms used can be seen for cases (II) and (III). The greatest savings in electric current consumption can be seen for case (IV), by lowering the current value and reducing data transmission.

To determine the average value of electric current consumed and the value of energy consumed by the end device, the Matlab Data Statistics Tool was used.

The average values of electric current consumed for cases I, II, III, and IV and for locations A, B, and C are shown in Table 2. As expected, the most electric current was consumed for the case without optimization algorithms (I). The lowest electric current consumption occurred for the case with implemented antenna optimization and temperature

prediction algorithms (IV). The value of the reduction in electric current consumption was similar to the study in the first series, for laboratory conditions. As the distance increased, the average value of electric current consumed increased by about 1 mA.

Table 2. Mean value of electric current.

		Mean Value of Electric Current [A]- Distance A	Mean Value of Electric Current [A]- Distance B	Mean Value of Electric Current [A]- Distance C
Cases	I	0.032	0.034	0.035
	II	0.031	0.033	0.033
	III	0.031	0.033	0.034
	IV	0.030	0.032	0.033

The electric energy consumed by the end device in 100 s for cases I, II, III, and IV and for locations A, B, and C is shown in Table 3.

Table 3. Electric energy consumption.

		Energy Consumption [W]-Distance A	Energy Consumption [W]-Distance B	Energy Consumption [W]-Distance C
Cases	I	0.160	0.170	0.175
	II	0.155	0.165	0.169
	III	0.156	0.166	0.171
	IV	0.151	0.161	0.166

The electric energy reduction values are similar to the savings obtained in the tests of the first series, in laboratory conditions. The reduction in electric energy consumption is between 1.2% for case II, 3.1% for case III, and 5.1% for case IV.

Limitations of the Proposed Method

Implementations of algorithms to reduce electric energy consumption by end devices were made for LoRaWAN. Implementation of the algorithms reducing electric energy consumption is performed on the end device (Raspberry Pi Pico platform: 2 MB flash memory and 264 kB SRAM memory). The microcontroller chips available today have enough processing power and memory resources to make the algorithms work on any end device. The algorithms were implemented on the smallest, weakest, and cheapest single-board microcomputer available.

6. Conclusions

From the available low-energy wireless communication networks, the LoRaWAN communication protocol was chosen for this study. To improve energy efficiency, two algorithms were proposed to optimize energy consumption. The first was based on the prediction of the value of the measured parameter, which was air temperature. The main task performed by the algorithm was to reduce the number of transmitted data packets from the end device to the gateway. The use of the prediction algorithm contributed to a reduction in energy consumption of only 1.12%. Significantly better results for reducing energy consumption were obtained by using an algorithm whose task was to optimize antenna gain. The use of this algorithm reduced energy consumption by 3.04% compared to without power limitation. Considering the standard high energy efficiency of LoRaWAN, this result can be considered satisfactory. The best results were achieved by combining the two proposed algorithms. Using both algorithms, a reduction in energy consumption

of up to 5.03% was achieved. The reduction in energy consumption when using both optimization algorithms is not the sum of the decreases in energy consumption when measuring each method separately, as shown in Figure 13. The combined operation of the two algorithms results in a reduction in the sent data, followed by an optimization of the antenna gain only at the remaining local maxima, as seen in Figure 11. In the second series of tests, the effectiveness of the proposed algorithms was checked with changing distance between the LoRa gateway and end device. This study showed no effect of power consumption optimization algorithms on the RSSI value. RSSI value changed only with the change in the distance between the LoRa gateway and end device. This study showed that there was no effect of electric power consumption optimization algorithms on the value of the rate of correctly received packets. With a short distance in the building (less than 24 m), the indicator had a value of almost 100%. In the second series of tests (tests with a change in the distance between the end device and the Lora gateway), similar reductions in electric energy consumption were obtained as in the test under laboratory conditions. It is worth emphasizing that the savings in electricity consumption were achieved without interfering with the communication protocol. There are about 500 million active IoT devices from the LPWAN group in the world (among others: Smart City, Smart Home, Smart Farming, Smart Factories, and Retail IoT) [43]. The 5% savings in electricity consumption equal about 500,000 € in savings per year. The number of connected IoT devices is growing by approximately 20% globally, in various industries. Energy prices are constantly growing, so the profit from the use of algorithms that reduce electricity consumption will also increase. The results of this study show that, in terms of the energy consumption of low-energy communication networks, it is possible to improve general solutions so as to make the operation of edge devices more efficient.

Author Contributions: Conceptualization, M.N., A.S. and G.S.; Methodology, M.N.; Software, A.S. and G.S.; Validation, R.R., J.S. and G.W.; Formal analysis, R.R., G.W. and M.N.; Writing—Original draft preparation, M.N., R.R., J.S. and G.W.; Writing—Review and editing, R.R., J.S., G.W. and M.N.; Supervision, R.R., G.W. and M.N.; Project administration, M.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Poznan University of Technology, project number 0311/SBAD/0708.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data needed for the testing experiments were collected by the authors.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

IoT	Internet of Things
LoRaWAN	Long Range Wide Area Network
LoRa	Long Range
LPWAN	Low Power Wide Area Network
SF	Spreading Factor
CF	Carrier Frequency
BW	Bandwidth
CR	Coding Rate
TP	Transmission Power
SNR	Signal to Noise Ratio
GSM	Global System for Mobile Communications
WiFi	Wireless Fidelity
FEC	Forward Error Correction
MAC	Medium Access Control
ADR	Adaptive Data Rate

ADTR	Adaptive Transmission Data Rate
SARA	Self-Adaptive Routing Algorithm
HAPS	High-Altitude Platform Stations
ISM	Industrial Scientific Medical
CSS	Chirp Spread Spectrum
ECB	Electronic Codebook
ITU	International Telecommunication Union
RSSI	Received Signal Strength Indication

References

- Adelantado, F.; Vilajosana, X.; Tuset-Peiro, P.; Martinez, B.; Melià-Seguí, J.; Watteyne, T. Understanding the Limits of LoRaWAN. *IEEE Commun. Mag.* **2017**, *55*, 34–40. [\[CrossRef\]](#)
- Butun, I.; Pereira, N.; Gidlund, M. Security Risk Analysis of LoRaWAN and Future Directions. *Future Internet* **2019**, *11*, 3. [\[CrossRef\]](#)
- AL-Hthloul, A.; Frikha, M. Analysis of Security Aspects in LoRaWAN. *J. Cyber Secur.* **2022**, *4*, 109–118. [\[CrossRef\]](#)
- Khalifeh, A.; Aldahdouh, K.; Alouneh, S. LoRaWAN Energy Optimization with Security Consideration. *Int. Arab. J. Inf. Technol.* **2021**, *18*, 476–483. [\[CrossRef\]](#)
- Antonowicz, A.; Derbis, P.; Nowak, M.; Urbaniak, A. Smart Meter in Voltage Control System of Power Network. In Proceedings of the 21th International Carpathian Control Conference (ICCC), High Tatras, Slovakia, 27–29 October 2020; pp. 1–6. [\[CrossRef\]](#)
- Bouguera, T.; Diouris, J.-F.; Chaillout, J.-J.; Jaouadi, R.; Andieux, G. Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN. *Sensors* **2018**, *18*, 2104. [\[CrossRef\]](#) [\[PubMed\]](#)
- Nurgaliyev, M.; Saymbetov, A.; Yashchyshyn, Y.; Kuttybay, N.; Tukymbekov, D. Prediction of energy consumption for LoRa based wireless sensors network. *Wirel. Netw.* **2020**, *26*, 3507–3520. [\[CrossRef\]](#)
- Sinha, S.R.; Wei, Y.; Hwang, S. A survey on LPWAN technology: LoRa and NB-IoT. *ICT Express* **2017**, *3*, 14–21. [\[CrossRef\]](#)
- LoRa Alliance Website, LoRa Technology. Available online: <https://loro-alliance.org/about-lorawan> (accessed on 15 October 2022).
- Cano-Ortega, A.; Sanchez-Sutil, F. Performance Optimization LoRa Network by Artificial Bee Colony Algorithm to Determination of the Load Profiles in Dwellings. *Energies* **2020**, *13*, 517. [\[CrossRef\]](#)
- Baumker, E.; Miguel Garcia, A.; Woias, P. Minimizing power consumption of LoRa and LoRaWAN for low-power wireless sensor nodes. *J. Phys. Conf. Ser.* **2019**, *1407*, 012092. [\[CrossRef\]](#)
- Conte, R.; Mahmud, S. LoRa-STAR: Optimizing Energy Consumption in LoRa Nodes for Precision Farming. In Proceedings of the 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 14 June–31 July 2021; pp. 183–188. [\[CrossRef\]](#)
- Augustin, A.; Yi, J.; Clausen, T. A study of LoRa: Long range low power networks for the Internet of Things. *Sensors* **2016**, *16*, 1466. [\[CrossRef\]](#) [\[PubMed\]](#)
- Devalal, S.; Karthikeyan, A. LoRa technology-an overview. In Proceedings of the 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 29–31 March 2018; pp. 284–290. [\[CrossRef\]](#)
- Zourmand, A.; Kun Hing, A.L.; Wai Hung, C.; AbdulRehman, M. Internet of Things (IoT) using LoRa technology. In Proceedings of the 2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Selangor, Malaysia, 29–29 June 2019; pp. 324–330. [\[CrossRef\]](#)
- Hamdi, R.; Baccour, E.; Erbad, A.; Qaraq, M.; Hamdi, M. LoRa-RL: Deep Reinforcement Learning for Resource Management in Hybrid Energy LoRa Wireless Networks. *IEEE Internet Things J.* **2022**, *9*, 6458–6476. [\[CrossRef\]](#)
- Noreen, U.; Bounceur, A.; Clavier, L. A study of LoRa low power and wide area network technology. In Proceedings of the 2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Fez, Morocco, 22–24 May 2017; pp. 1–6. [\[CrossRef\]](#)
- Hou, Y.; Liu, Z.; Sun, D. A novel MAC protocol exploiting concurrent transmissions for massive LoRa connectivity. *J. Commun. Netw.* **2020**, *22*, 108–117. [\[CrossRef\]](#)
- Harada, H.; Mizutani, K.; Fujiwara, J.; Mochizuki, K.; Obata, K.; Okumura, R. IEEE 802.15.4g Based Wi-SUN Communication Systems. *IEICE Trans. Commun.* **2017**, *E100.B*, 1032–1043. [\[CrossRef\]](#)
- Kaven, S.; Bornholdt, L.; Skwarek, V. Authentication by RSSI-Position Based Localization in a LoRa LPWAN. In Proceedings of the 2020 6th IEEE Congress on Information Science and Technology (CiSt), Agadir-Essaouira, Morocco, 5–12 June 2021; pp. 448–454. [\[CrossRef\]](#)
- Nolan, K.E.; Guibene, W.; Kelly, M.Y. An evaluation of low power wide area network technologies for the Internet of Things. In Proceedings of the IEEE International of Wireless Communications and Mobile Computing Conference (IWCMC), Paphos, Cyprus, 5–9 September 2016; pp. 440–444. [\[CrossRef\]](#)
- Kufakunesu, R.; Hancke, G.; Abu-Mahfouz, A. A Fuzzy-Logic Based Adaptive Data Rate Scheme for Energy-Efficient LoRaWAN Communication. *J. Sens. Actuator Netw.* **2022**, *11*, 65. [\[CrossRef\]](#)

23. Mendes, B.; Correia, N.; Passos, D. On the Optimization of LoRaWAN Gateway Placement in Wide Area Monitoring Systems. In *Internet of Things; IoT through a Multi-Disciplinary Perspective*. IFIPIoT 2022. IFIP Advances in Information and Communication Technology; Camarinha-Matos, L.M., Ribeiro, L., Strous, L., Eds.; Springer: Cham, Switzerland, 2022; Volume 665. [CrossRef]
24. Dandachi, G.; Hadjadj-Aoul, Y. A Frequency-Based Intelligent Slicing in LoRaWAN with Admission Control Aspects. In *Proceedings of the International Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*, Montreal, Canada, 24–28 October 2022; pp. 189–196. Available online: <https://dl.acm.org/doi/abs/10.1145/3551659.3559055> (accessed on 7 November 2022).
25. Zahid, N.; Sodhro, A.H.; Kamboh, U.R.; Alkhayyat, A.; Wang, L. AI-driven adaptive reliable and sustainable approach for internet of things enabled healthcare system. *Math. Biosci. Eng.* **2022**, *19*, 3953–3971. [CrossRef] [PubMed]
26. Maudet, S.; Andrieux, G.; Chevillon, R.; Diouris, J.F. Refined node energy consumption modeling in a LoRaWAN network. *Sensors* **2021**, *21*, 6398. [CrossRef] [PubMed]
27. Nowak, M.; Derbis, P.; Kurowski, K.; Różycki, R.; Waligóra, G. LPWAN Networks for Energy Meters Reading and Monitoring Power Supply Network in Intelligent Buildings. *Energies* **2021**, *14*, 7924. [CrossRef]
28. Semtech White Paper, Real-world LoRaWAN™ Network Capacity for Electrical Metering Applications, Semtech Corporation. Available online: https://info.semtech.com/network_capacity_white_paper_download (accessed on 15 October 2022).
29. Giambene, G.; Korre, K. LoRa-based System for IoT Applications via HAPS in Remote Areas. In *Proceedings of the 2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Croatia, 22–24 September 2022; pp. 1–6.
30. Pasolini, G. On the LoRa Chirp Spread Spectrum Modulation: Signal Properties and Their Impact on Transmitter and Receiver Architectures. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 357–369. [CrossRef]
31. IC880A-SPI Technical Documentation. Available online: <https://wireless-solutions.de/downloadfile/ic880a-spi-documents/> (accessed on 15 October 2022).
32. BeagleBone Black Technical Documentation. Available online: https://cdn-shop.adafruit.com/datasheets/BBB_SRM.pdf (accessed on 15 October 2022).
33. LoRa Alliance Technical Committee, LoRaWAN™ 1.1 Specification, Version 1.1, LoRa Alliance. Available online: https://loralliance.org/wp-content/uploads/2020/11/lorawantm_specification_v1.1.pdf (accessed on 15 October 2022).
34. Raspberry Pi Pico Technical Documentation. Available online: <https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf> (accessed on 15 October 2022).
35. Hoperf RFM95W Technical Documentation. Available online: <https://www.hoperf.com/data/upload/portal/20190801/RFM95W-V2.0.pdf> (accessed on 15 October 2022).
36. Adafruit SHT40 Technical Documentation. Available online: <https://www.mouser.com/pdfDocs/4885SensirionSHT40Sensor.pdf> (accessed on 15 October 2022).
37. RapidJSON Library Technical Documentation. Available online: <https://rapidjson.org/> (accessed on 15 October 2022).
38. RIGOL DM3051 Technical Documentation. Available online: <https://www.rigolna.com/products/digital-multimeters/dm3000/> (accessed on 15 October 2022).
39. Ywrobot MB-V2 Technical Documentation. Available online: https://static.rapidonline.com/pdf/73-4538_v1.pdf (accessed on 15 October 2022).
40. Rigol instruments and MATLAB. Available online: <https://ch.mathworks.com/products/instrument/supported/rigol-technologies.html> (accessed on 10 December 2022).
41. Driver Registry. IVI Foundation. Available online: https://www.ivifoundation.org/registered_drivers/driver_registry.aspx (accessed on 10 December 2022).
42. TTN—The Things Network, TTS—The Things Stack—Open Source Infrastructure Aiming at Providing a Free LoRaWAN Network Cover. Available online: <https://www.thethingsnetwork.org> (accessed on 10 December 2022).
43. Number Connected IoT Devices. Available online: <https://iot-analytics.com/number-connected-iot-devices> (accessed on 10 December 2022).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.