Development of an Energy Management System in Buildings based on Edge Computing and Machine Learning

Jorge Pedro
PPGEEC — UFC
Sobral, Brazil
Beira, moçambique

Wendley Souza da Silva
PPGEEC — UFC
Sobral, Brazil

Geraldo Eufrazio Martins Júnior
PPGEEC — UFC
Sobral, Brazil

ABSTRACT

The exponential increase in demand for electricity, especially in urban centers, has put the sustainability and efficiency of energy supply systems in check [1]. This paper presents the development and experimental validation of an energy management and optimization system in Buildings Based on Edge Computing, using the ESP32-S3 microcontroller and embedded Machine Learning (ML) algorithms. The system consists of a central processing unit and distributed sensing and actuation modules, communicating via the ESP-NOW wireless protocol without the need for an external router. The voltage and current sensors, based on the IC ATM90E36 with resistive splitters and ZMCT103C current transformers, were validated in the laboratory with R^2 greater than 0.9998 and average errors of ±0.17% and ±0.30%, respectively — performance higher in current than reported by [2]. ESP-NOW communication at 20 meters with masonry obstacles achieved a 99.5% success rate and 5.8 ms average latency (250 ms interval). The Random Forest Regressor model, trained on 718 hourly samples of 30 days of continuous operation, achieved MAE of 70.71 W (22.7% reduction compared to baseline) and was integrated directly into the ESP32-S3 firmware with an inference time of 20 ms and no dependence on external connectivity.

General Terms

Edge Computing, Machine Learning, Internet of Things, Embedded Systems, Energy Efficiency, FreeRTOS.

Keywords

ESP32-S3; ESP-NOW; ATM90E36; Random Forest; Building Energy Management; Embedded Machine Learning; ZMCT103C.

1. INTRODUCTION

The exponential increase in demand for electricity, especially in urban centers, has put the sustainability and efficiency of energy supply systems in check [1]. Since the Industrial Revolution, modern society's dependence on electricity to supply industrial, commercial, and domestic activities has intensified, resulting in a growing consumption that challenges the capabilities of traditional infrastructures. This scenario is aggravated by the need to reduce the environmental impacts associated with energy generation, aligning with global sustainability goals, such as those established by the Paris Agreement [3]. Thus, the search for strategies that promote the efficient use of energy has become a priority both to mitigate the effects of climate change and to balance the relationship between generation and consumption.

Modernizing energy infrastructure through innovative technologies is essential to address these challenges. In recent years, advances in the Internet of Things (IoT) and Artificial Intelligence (AI) have enabled the development of intelligent systems for monitoring and controlling energy consumption in real time[4]. Such systems allow for the collection and analysis of detailed data, offering solutions such as dynamic load adjustment and prediction of usage patterns, which can significantly reduce energy waste.

In this context, energy efficiency stands out as a strategy applicable to the generation, distribution and end use of electricity. [5], improvements in consumer systems — such as replacing inefficient equipment or implementing automated controls — can optimize energy performance without harming operations.

The integration of edge computing with low-cost microcontrollers, such as the ESP32-S3, and ML algorithms further expands these possibilities, reducing latency and bandwidth consumption when processing data locally[6]. Edge computing moves processing, storage, and analytics operations that would traditionally be performed in the cloud to the edge of the network, close to end devices or data sources [7] Despite these innovations, gaps persist in the development of affordable and scalable solutions that combine high metrological accuracy, experimental validation, and adaptation to real environments.

In view of this panorama, this research proposes the development of a building energy management and optimization system based on Edge Computing, using ML and the ESP32-S3 microcontroller. The objective is to design a solution that monitors and allows adjustment of energy consumption in real time, adapting to the computational limitations of the embedded hardware. The relevance of this work lies in its contribution to building energy efficiency, offering a practical and low-cost approach that can be replicated in various urban contexts, especially in scenarios with limited telecommunications infrastructure, such as those identified in the literature review [8].

The central question of the research is: How can an Edge Computing-based system, using ML in the ESP32-S3 microcontroller, manage and optimize energy consumption in Buildings? To answer it, the overall goal is to develop and experimentally validate a building energy management and optimization system based on Edge Computing with ML and ESP32-S3

2. RELATED WORK

This section looks at the state of the art in building energy

efficiency with a focus on the integration of IoT and AI. The research was conducted in IEEE Xplore, ResearchGate and Google Scholar, using terms such as 'IoT', 'energy efficiency', 'building automation' and 'artificial intelligence applied to energy efficiency'. Articles published between 2015 and 2024 were considered, prioritizing experimental studies with technical relevance to the use of IoT and AI in embedded systems and practical impact on reducing energy consumption.

2.1 IoT Applied to Energy Efficiency in Buildings

Population growth and urbanization intensify the demand for electricity, raising concerns about the sustainability of the energy sector [1]. IoT plays a crucial role in enabling smart devices for energy monitoring and control[9]. developed an IoT socket for consumption monitoring, demonstrating usefulness in contexts of distributed generation and non-linear loads, but without integration with automated controls. [10] advanced by proposing a system based on ESP32 and PZEM-004T sensors with the Blynk platform for real-time monitoring of voltage, current and power. Although effective in automation, the work does not quantify energy reduction, limiting its practical applicability.

[11] combined IoT with prediction via ML algorithms (ARIMA, Prophet and LSTM), highlighting LSTM as the most accurate for estimating future consumption. The inclusion of Android application for visualization and alerts extends user control, but the reliance on cloud connectivity can be disadvantageous in limited infrastructure scenarios. [12] explored innovative approach with IoT and Digital Twins, creating virtual representations of devices for detailed analysis of energy use. The results indicated savings of up to 25% at peak times; however, the complexity of implementation may restrict its large-scale adoption. [8] proposed a low-cost system with ESP32, Blynk and notifications via WhatsApp, focused on regions with electricity scarcity such as the Gaza Strip — accurate monitoring with reduced waste, but with limited predictive functionalities.

2.2 Artificial Intelligence Applied to Energy Efficiency

[4] developed an open-source Edge Computing platform at the University of Colima, using sensors for voltage, current, and active power monitoring. By processing data locally with SQLite and MQTT, the system reduces latency and improves security compared to cloud-only architectures. Its scalability and open-source nature make it adaptable, but laboratory testing lacks validation in real-world environments. [13] proposed a management system for smart homes based on deep learning, combining 1D Convolutional Neural Network (1D-DCNN), bidirectional long-term memory (BLSTM) and reinforcement learning (QL). Validated by simulations, the BLSTM model outperformed traditional neural networks in long-term load prediction, optimizing consumption and comfort. The high computational complexity, however, may be unfeasible for microcontrollers such as the ESP32-S3.

2.3 Synthesis and Contributions to the Project

The review reveals that IoT and AI offer complementary solutions for building energy efficiency: while IoT systems such as those of [10] e [8] highlight the feasibility of ESP32 in low-cost monitoring, predictive approaches such as [11] e [13] expand the potential of proactive management. Comparatively, Edge Computing [14] overcomes cloud latency limitations, but few studies integrate these technologies into experimentally validated embedded systems. Gaps persist in quantifying energy impact and adapting to real scenarios, especially in contexts with limited infrastructure. This work is positioned by combining the ESP32-S3 with machine learning at the edge, seeking to fill these gaps with an affordable solution, tested in the lab and in a building environment.

3. METHODOLOGY

The Design Science Research (DSR) methodology was adopted, materialized in a functional prototype, according to the research strategy proposed by [15]. DSR is particularly suitable for research aimed at the development of functional prototypes, systematically integrating the stages of problem identification, solution design, implementation and performance evaluation. The research has an experimental nature and a mixed approach, combining quantitative methods in the analysis of performance and validation of results, with qualitative methods in the definition of requirements and interpretation of the results obtained. The methodology is structured in six stages: Bibliographic Survey, Hardware Prototype Development, Software Development, Laboratory Tests and Analysis of Results as recommended by [16].

3.1 Architecture and System Requirements

The system is designed to meet functional requirements — real-time monitoring of voltage, current, and power; automatic and remote control of electrical loads; efficient communication between central and modules; data logging for analysis and non-functional requirements — low latency in ML inference; low power consumption; reliability in communication without additional external devices; scalability; and response times that support real-time monitoring.

The architecture integrates local processing, distributed communication, and user interaction via the ESP-NOW network. The Central Unit is responsible for the local processing of the data, execution of inferences from the ML model and graphical interface via Nextion display. The central unit connects to the modules via ESP-NOW, acting as a gateway to external Wi-Fi networks when available.

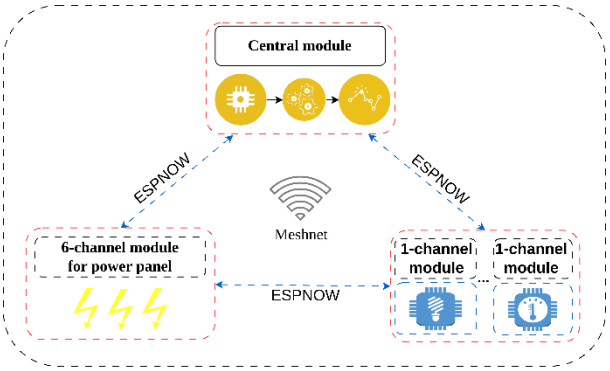

Figure 1. Architecture of the proposed system

3.2 Hardware Development

The circuits were designed in Proteus software with pre-3D visualization and validation on ESP32-DevKit boards prior to final manufacturing. The Center integrates: ESP32S3WROOM1N16R2 (dual-core LX7, 240 MHz, Wi-Fi, Bluetooth 5.0); DS3231MZ (temperature-compensated RTC, ±2 ppm, I²C interface); MicroSD (SPI) slot for local CSV storage; ESP-PSRAM64H (64 Mbit, pseudo-SRAM for intensive inferences); W25Q512JVEIQ (512 Mbit Flash, Quad-SPI, 133 MHz, 100,000 cycles); TDK MMICT5848 microphone; and USB Type-C connector for programming.

The power supply circuit uses the LTC4054LES5 for 4.2V lithium-ion battery charging, DW01A-G protection, TLV76733 regulator (LDO 3.3V, 1A, 300mV dropout), and Schottky 1N5819HW7 diodes in Diode ORing configuration for automatic management between battery and external source. Figure 2 presents the layout of the proposed board.

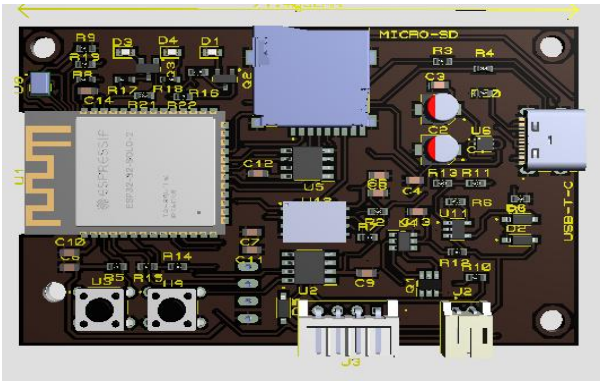

Figure 2. Central control board

The 6-Channel Module is based on two ATM90E36 ICs (U1 and U4) connected to the ESP32-S3 via SPI. The current readout circuit uses the CST3061A transformer (5A/5mA) with RC filters (fc = 50 Hz) for noise mitigation, replicated for all six channels. The voltage readout circuit uses splitter with 7 x 240 kΩ resistors in series and 1 x 10 kΩ resistor, reducing 220 V AC to 0–3.3 V (power dissipated by 240 kΩ resistor: 4.07 mW). The five 5 V relays are driven by NPN transistors in Darlington configuration—multiplied gains, lower current demand of the ESP32-S3 (BOYLESTAD and NASHELSKY, 2013)—with protection diodes for voltage surges during switching. The source uses HLK-PM03 (220 V → 5 V, >85% efficiency) and AMS1117-3.3 (1.2 V dropout, 1 A) with two isolating diodes in Diode ORing configuration. The board design can be seen in Figure 3.

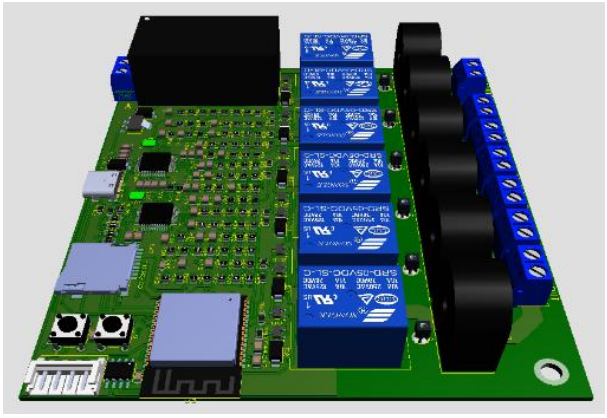

Figure 3. 6-channel readout board

The 1-channel module, depicted in Figure 4, is based on the ESP8266-12E microcontroller, powered at 3.3 V via an AMS1117-3.3 regulator. For power monitoring, the HLW8032 integrated circuit is employed, communicating via UART at 115200 bps. The actuation system consists of a 5V relay driven by a Darlington pair of BC817-40 NPN transistors connected to GPIO2, including a D2 freewheeling diode for protection. Environmental sensing is performed by a DHT22 sensor on GPIO4, supported by power supply filter capacitors. Following the generation of Gerber files, the PCBs were manually assembled using a soldering station, followed by rigorous continuity and power-up tests prior to system integration.

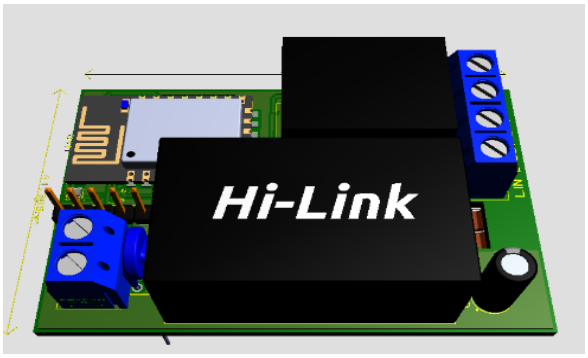

Figure 4. Single-channel readout board

3.3 Firmware with Layered Architecture and FreeRTOS

The firmware was developed in Visual Studio Code with the Arduino platform for ESP32, following the Layered Architecture standard [17]: Layer 1 — Hardware abstraction (SPI, I²C, UART drivers); Layer 2 — Signal processing (calibration, calculation of electrical quantities, time aggregation); Layer 3 — Communication (ESP-NOW, UART protocol for Nextion display); Layer 4 – Persistence (NVS for calibration parameters, FAT32 for MicroSD storage). Each functional module is implemented as an independent task in FreeRTOS, allowing concurrent execution.

The measurement module firmware performs sequential reading of the six channels per cycle (optimizing acquisition time), stores the data in a CSV file on the MicroSD, and transmits the consolidated readings in a single package via ESP-NOW at each configurable interval of 1 second or 1 minute. Calibration parameters are stored in NVS and automatically loaded on reboots. The center receives the packages of the modules, executes the ML model for consumption prediction and updates the graphical interface on the Nextion display. The interface comprises seven screens as described in the UI section.

3.4 Testing and calibration of measurement modules

The experiments were conducted in a real operating environment, on the premises of the UFC-Federal University of Ceará, specifically in the electrotechnical laboratory and in classrooms of the Mocambinho-Sobral campus. The choice of this environment was aimed at technical conditions of the laboratory for calibration and classrooms to simulate realistic conditions for the implementation of the system, considering the typical constructive characteristics of buildings.

The Hikari HM-2030 digital multimeter is a True RMS type instrument, suitable for voltage and current measurements in alternating and direct current, even in conditions where the waveforms are not perfectly sinusoidal. For alternating voltage measurements in the ranges from 2 V to 200 V, the equipment has an accuracy of ±(0.5% + 3 digits) and decimal values, an essential characteristic for rigorous linearity analysis[18].

The voltage for the test of the system was supplied by a variac connected to the electrical network, allowing the continuous variation of the output voltage from 0 V to values higher than the nominal voltage of the network. The use of variac allowed the precise definition of test points, the minimization of transients and the controlled simulation of real operating conditions of the electrical network.

3.5 Calibration of the measurement module

The calibration and validation of voltage and current sensors are fundamental steps to ensure the reliability, accuracy and repeatability of the measurements performed by the proposed system. The calibration procedure was conducted in accordance with the manufacturer's recommendations and using the equations specified in the datasheet for the calculation of the new gain values based on reference measurements [19]

3.5.1 Voltage Sensor Calibration

In the process of calibrating the voltage sensor, initially, the ATM90E36's voltage gain recorders were configured with the default values provided by the manufacturer, corresponding to unit gain. Then, using the variac, an alternating voltage of 150V RMS was applied. The voltage was gradually adjusted and continuously monitored by the Hikari HM-2030 multimeter, connected in parallel to the system under test.

After a stabilization period of approximately 30 seconds, the voltage values measured by the ATM90E36 and the reference multimeter were recorded. The new voltage gain was calculated according to the equation recommended by the manufacturer:

$$NGV = \left(\frac{Reference\ Voltage}{Measured\ Voltage}\right) \times Existing\ Gain$$

Where:

NGV – New Gain for Voltage

The calculated value was recorded in the corresponding ATM90E36 register, followed by the system restart for effective application of the calibration.

3.6 Determination of Linearity

The validation of the linearity of the voltage sensor was performed using variac for precise definition of the test points. During this procedure, it was decided to restrict the tests to the range of 100 V to 200 V RMS, in increments of 20 V.

This choice was motivated by the metrological characteristics of the reference multimeter. In the range of 2 V to 200 V, the HM-2030 maintains an accuracy of ±0.5%, in addition to having resolution to decimal places. For voltages above 200 V, the instrument automatically moves to the 750 V range, in which accuracy is reduced to ±1.0%, and readings no longer show decimal values, compromising the resolution required for a detailed linearity analysis.

For each voltage point defined as reference (100 V, 120 V, 140 V, 160 V, 180 V and 200 V), 10 consecutive measurements were performed, with an interval of 1 second between readings. Before each set of measurements, a 30-second stabilization period was waited. The data were acquired by the ESP32-S3 via SPI interface, with a sampling rate of 1 kHz, and stored for later statistical analysis.

3.7 Current Sensor Calibration

The calibration of the current sensor followed a methodology analogous to that used for voltage. A series circuit was assembled containing the current transformer, the ATM90E36 and a controlled resistive load. The Hikari HM-2030 multimeter has been connected in series for reference current measurement.

The initial calibration current was established at 1.5A RMS, intermediate value of the operating range, obtained by adjusting the output voltage of the variac. After thermal stabilization of the circuit (approximately 60 seconds), the ATM90E36 and multimeter readings were collected, and the new current gain was calculated by the equation:

$$NGC = \left(\frac{Reference\ Value}{Measured\ Value}\right) \times Existing\ Gain$$

Where:

NGC – New Gain for Current

3.8 Determination of Current Sensor Linearity

After the initial calibration, the linearity of the current sensor was validated in the operating range between 0 A and 5 A. This range was selected because it is representative of the consumption profile associated with low-power residential loads, aligning the conditions of the experimental test and typical application conditions of the proposed system.

The different current levels were obtained by adjusting the output voltage of the variac, associated with the controlled modification of the load configuration. For this purpose, a bench composed of ten incandescent lamps was used, which could be activated individually or in different combinations, allowing the generation of multiple measurement points distributed along the range of interest.

For each current value established, ten consecutive measurements were performed, with an interval of two seconds between readings, after a previous stabilization period of thirty seconds. The interval adopted between the measurements was intended to ensure the stabilization of the readings and minimize the influence of thermal effects associated with the heating of the lamp filaments.

Throughout the experimental procedure, the Hikari HM-2030 digital multimeter remained connected in series to the circuit, providing the reference current values. The acquired data were later processed to obtain the linearity line and enabling the quantitative evaluation of the linearity of the sensor.

3.9 Validation of Communication between Modules

The communication between the measurement modules and the processing center is a critical element for the operation of the proposed system, directly influencing the integrity of the data collected and the responsiveness of the anomaly detection system. This section presents the methodology used in the validation tests of communication via the ESP-NOW protocol.

The devices were positioned in two non-adjacent classrooms, separated by a third intermediate classroom. This configuration resulted in an approximate distance of 20 meters between the transmitter module and the receiver, with the presence of masonry walls as physical obstacles to the radio frequency signal. This condition represents a challenging scenario for wireless communication, characterized by significant signal attenuation due to propagation across multiple physical barriers.

3.10 Transmission Integrity Assessment

The evaluation of the integrity of the transmission was carried out through systematic tests that consisted of sending 1,000 consecutive packets at different transmission intervals. The objective of this approach was to characterize the relationship between transmission rate and system reliability, identifying the operational limits of the ESP-NOW protocol in the test environment.

Five distinct intervals between consecutive transmissions were tested: 50 ms, 100 ms, 250 ms, 500 ms, and 1,000 ms. For each interval, the number of packets actually received by the processing center was recorded, allowing the calculation of the success rate and the packet loss rate.

3.11 Characterization of Communication Latency

In addition to the integrity tests, a detailed characterization of the communication latency was carried out under different system load conditions. Latency was defined as the time interval between the sending of the packet by the transmitter module and its reception confirmed by the processing center.

The methodology employed consisted of the systematic measurement of the propagation time for each of the 1,000 packets transmitted in each test condition. For each interval between transmissions, the mean, minimum, maximum latency and standard deviation were calculated, allowing complete statistical characterization of the temporal behavior of the system.

3.12 microSD Card Storage

After reading the six channels, the module saves all the data in a CSV file on the microSD card. Upon starting, the system checks for a file with a previously defined name on the card. If the file exists, the system continues to write to it, adding new records at the end. Otherwise, a new file is created automatically.

Each record includes the precise date and time and data of all six channels: channel identification and its respective measured parameters. This strategy allows continuity in data collection even after system reboots, maintaining a complete and organized history.

To prevent data loss in the event of a power outage, the firmware uses a write buffer and periodically synchronizes with the card. This minimizes the amount of data lost if the system shuts down abruptly.

3.13 User Interface

To allow the user to interact with the system, a full graphical interface has been developed on the 4.3-inch Nextion Edge Series display. This display was chosen for its easy integration with the ESP32 and for running the graphical interface independently, reducing the microcontroller processing load. The interface was created using the Nextion Editor software and communicates with the ESP32 through a serial port. Six screens were developed, each with a specific function: Main Menu figure 1, channel control screen figure 2, General Electrical Parameters screen figure 3, Channel Details screen figure 4, Settings screen figure 5, and the Anomaly Analysis screen figure 6.

Figure 5. Access menu screen

Figure 6. Channel control information screen

Figure 7. General Electrical Parameters

Figure 8. Details by channel

Figure 9. System configuration

Figure 10. Anomaly Analysis

3.14 ML Training

This section presents the procedures adopted for predictive modeling of energy consumption from the data collected by the developed system. The objective was to verify the feasibility of hourly forecasting of residential consumption, compare regression approaches and evaluate whether the database produced by the prototype presents sufficient consistency for the application of machine learning techniques.

3.14.1 Dataset Characterization

The dataset comprises 86,393 records collected over 30 days of continuous operation, with a sampling interval of 30 seconds. Figure X shows the data collection system installed to gather the energy consumption data. Although the system supports up to six simultaneous channels, the collection was limited to two channels due to the configuration of the monitored electrical installation, which has only two main distribution circuits in the electrical panel.

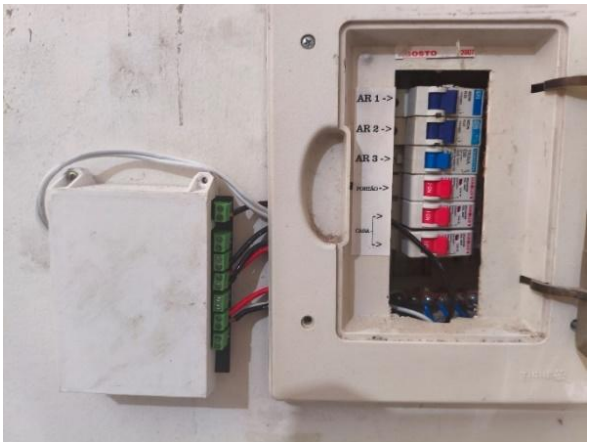

Figure 11. Data collection system installed for energy consumption monitoring.

Channel 3 showed greater variability (mean of 448.9 W; standard deviation of 363.1 W), indicating the presence of non-continuously driven loads. Channel 6, on the other hand, showed more stable behavior (mean of 270.9 W; standard deviation of 99.0 W), compatible with permanent or frequently used loads.

In the period analyzed, the total consumption was 518.20 kWh, unevenly distributed between the two channels (62.4% in Channel 3 and 37.6% in Channel 6). This difference not only characterizes the circuits but also directly influences the difficulty of the prediction problem, since greater variability tends to increase the expected error.

3.15 Preparation and Pre-processing

The raw data were aggregated temporally from 30 seconds to one-hour intervals, calculating the arithmetic mean of the power in each window. This choice is justified for three reasons: (i) hourly aggregation attenuates high-frequency noise, (ii) short-term oscillations with no relevance to the adopted forecast horizon, and (iii) aligns the granularity of the data with the desired prediction interval (consumption of the next hour). After aggregation, the resulting dataset had 718 hourly samples. Seven attributes were constructed to characterize each sample: Total Power, Channel 1 Power, Channel 6 Power, Frequency, Time of Day, Day of the Week, Weekend.

The target variable was defined as the total power of the immediately following hour, configuring the problem as supervised regression with a horizon of one hour ahead.

3.16 Data Division and Validation Strategy

In time-series problems, shuffling the data before the training/test split introduces future information leakage, compromising the validity of the metrics obtained [20]. For this reason, the division strictly respected chronological order: the first 70% of the records (502 samples) were reserved for training and validation, and the remaining 30% (216 samples) for final testing.

The test suite remained isolated throughout the development process, including pre-processing and hyperparameter optimization, being used exclusively in the final evaluation of the models, preventing any information from the test from influencing the modeling decisions as recommended by KAPOOR and NARAYANAN, (2023). Once the strategy for dividing and validating the data was defined, the next step consisted of the implementation and comparison of the regression models, described in the following section

3.17 Machine Learning Models Deployed

Three regression models were implemented in order to identify the most appropriate approach for integration into the developed prototype. The final choice was not pre-established, being defined based on the performance obtained in the isolated test set (216 samples). The comparison followed increasing order of complexity: Baseline by Historical Average, Random Forest Regressor and Gradient Boosting Regressor. This progression allowed us to evaluate whether the increase in complexity would result in a measurable performance gain for the dataset of 718 hourly samples.

The model selection prioritized the MAE and MAPE metrics in the test set, for their direct interpretation in the context of energy consumption — error in Watts and percentage, respectively. The consistency between the results of the validation and the final test was also evaluated — divergences greater than 15% in the main metrics were interpreted as an indication of overfit, disqualifying the model for application in the prototype regardless of the absolute performance obtained. Models with negative R^2 were also discarded, as they indicated lower performance than the baseline itself.

The model selected was the one that presented the best balance between predictive accuracy and stability between the evaluation phases, considering that the ultimate goal is integration in real operating conditions, and not just performance in a controlled environment.

3.18 Evaluation Metrics

The four metrics used in the comparison of the models are presented below, with the respective equations and the form of interpretation applied in this work. In all expressions, $\hat{y}i$ represents the value predicted by the model, yi the actual

observed value, and n the number of samples evaluated.

3.18.1 MAE — Mean Absolute Error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

Expressed in Watts, the MAE represents the expected average error in each prediction. Because it is calculated on an absolute scale, it allows the typical deviation of the model to be directly interpreted in the context of the monitored system — a 70 W MAE over an average consumption of 720 W has a different operational meaning than the same error over a 100 W load.

3.18.2 RMSE — Root Mean Squared Error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

By squaring errors before averaging, RMSE penalizes point deviations more severely than MAE. In this work, it was used in conjunction with MAE to identify models with unstable behavior at Channel 3 peak hours — a significant difference between RMSE and MAE indicates the occurrence of isolated errors of great magnitude.

3.18.3 MAPE — Mean Absolute Percentage Error

$$MAPE = \frac{1}{n}\sum_{1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100$$

By normalizing the error by the actual value, MAPE allows you to compare the performance of the models regardless of the scale of consumption of each channel. Given that Channel 3 has an average of 448.9 W and Channel 6 has an average of 270.9 W, the percentage analysis avoids that larger absolute errors in Channel 3 are interpreted as necessarily lower performance.

3.18.4 R^2 — Coefficient of Determination

$$R^2 = \frac{\sum_{n}^{1}(\hat{y}_i - y_i)^2}{\sum_{n}^{1}(y_i - \bar{y})^2}$$

Where $\bar{y}$ represents the average of the actual values. The R^2 expresses the proportion of the variance of consumption explained by the model: values close to 1 indicate good fit, values close to 0 indicate performance equivalent to the simple average, and negative values indicate performance below the baseline. As set forth in section 6.1.4, models with negative R^2 were automatically disqualified. Because it is sensitive to the autocorrelation present in time series, R^2 was not used as an isolated selection criterion.

3.19 Integrating the Model with the Prototype Firmware

After selecting the final model, the integration with the hardware followed an embedded inference approach: the model is trained offline on a computer and later converted into a C++ header file (.h), embedded directly in the firmware of the ESP32-S3 microcontroller without the need for external communication or dedicated server.

The integration process comprised three stages. In the first, the model trained in Python is converted into a modelo_consumo.h file, containing the model structure in a format compatible with the ESP32 environment.

In the second step, the generated .h file was included in the main firmware code, providing the prever_consumo() function, which receives a vector of seven features in real time and returns the consumption estimate for the next 90-second horizon (3 intervals of 30 seconds). The features used in the embedded inference correspond to a subset of those built in the training phase, limited to the quantities available in real time on the device: current power of the two channels (P1C, P2C), total power, voltage and current of each channel (V1C, I1C, V2C, I2C), moving averages of 3 periods and time of day.

In the third step, the firmware was configured to run the prediction autonomously each acquisition cycle, without dependence on connectivity or external processing. This architecture ensures that the prototype maintains predictive capacity even in situations of network failure, making the system more robust for continuous residential application.

The methodological path described in this chapter reflects the decisions taken throughout the development of the system, which are not always linear and often conditioned by the real characteristics of the monitored facility and the limitations of the available resources. The calibration of the sensors in a laboratory environment, the communication tests between modules in real obstruction conditions, and the collection of data during 30 days of continuous operation were not just formal steps in a process, they were the moments when the system was confronted with reality and adjusted accordingly.

The integration of the machine learning model into the ESP32-S3 firmware represents the point of convergence between all these steps: the data collected by the hardware, processed and validated throughout the methodology, fed into a model that now operates autonomously in the prototype itself. It is this complete cycle — from measurement to forecasting, without dependence on external infrastructure — that the following chapter proposes to evaluate, presenting the results obtained and discussing the extent to which the developed system corresponds to the defined objectives and requirements

4. RESULTS AND DISCUSSION

This chapter presents the results of laboratory tests performed for system validation. The analysis is divided into three stages: validation of the accuracy of the sensors (voltage and current), validation of communication between modules and validation of the Machine Learning model.

4.1 Voltage Reading

Validating the sensors is critical to ensuring the reliability of measurements, as the quality of the input data directly influences the ML model's ability to make correct predictions. Table 1 presents the results of the linearity tests of the voltage sensors with measurements performed in the variac. For each reference voltage point, 10 consecutive measurements were taken, calculating the average. The expanded uncertainties are between ±0.04 V and ±0.08 V. Graph 1 compares the reference values with the voltages measured in the six channels.

Table 1- Values Obtained for Linearity Validation of Voltage Sensors

V ref (V)	Channel 1	Channel 2	Channel 3 (V)	Channel 4	Channel 5 (V)	Channel 6 (V)
100,0	100,25 ±0,04	100,10 ±0,05	100,00 ±0,04	100,00 ±0,05	99,90 ±0,06	99,75 ±0,07
120,2	120,60 ±0,05	120,35 ±0,06	120,20 ±0,05	120,20 ±0,06	120,05 ±0,07	119,80 ±0,08

122,6	123,10 ±0,06	122,80 ±0,05	122,65 ±0,06	122,65 ±0,05	122,50 ±0,07	122,20 ±0,08
140,1	140,80 ±0,07	140,45 ±0,06	140,25 ±0,07	140,25 ±0,06	140,05 ±0,08	139,70 ±0,07
160,5	161,30 ±0,08	160,90 ±0,07	160,65 ±0,08	160,65 ±0,07	160,40 ±0,08	160,00 ±0,08
180,1	181,00 ±0,08	180,55 ±0,07	180,30 ±0,08	180,30 ±0,07	180,05 ±0,08	179,60 ±0,08
199,1	200,10 ±0,08	199,60 ±0,07	199,35 ±0,08	199,35 ±0,07	199,10 ±0,08	198,50 ±0,08

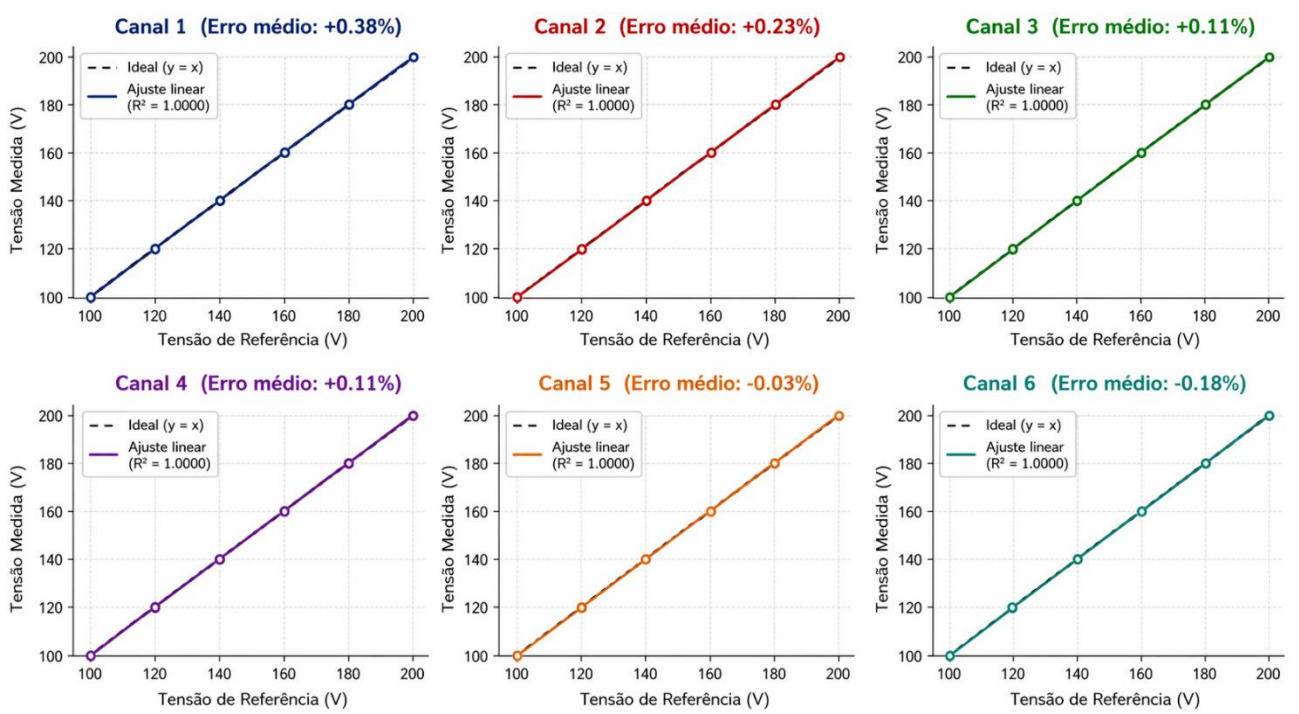

Figure 12. - Linearity of voltage sensors: 6-channel curves vs. 100–200 V reference.

Table 2- Voltage Sensor Performance Indicators

Channel	R^2	Average error (%)	Max. error (%)	Average Uncertainty (V)	ENL (%)
1	0,9999	+0,38	+0,50	±0,06	0,59
2	0,9999	+0,23	+0,35	±0,06	0,42
3	0,9999	+0,11	+0,20	±0,06	0,31
4	0,9999	+0,11	+0,20	±0,06	0,31
5	0,9999	−0,03	+0,10	±0,07	0,20
6	0,9999	−0,18	−0,10	±0,07	0,48

The data demonstrate adequate linear behavior along the range of 100 V to 200 V. Linear regression by the least squares method resulted in coefficients of determination R^2 higher than 0.9999 for all channels, indicating that more than 99.99% of the observed variation follows a linear relationship. The nonlinearity error (ENL) is between 0.20% and 0.59%, confirming the suitability of the resistive dividers for this application.

Comparative analysis between channels reveals systematic differences. Channel 1 shows consistent overestimation (mean error of +0.38%), while Channel 6 exhibits underestimation (−0.18%). Channels 3 and 4 show greater proximity to the reference values, with errors of less than ±0.20%. Inter-channel dispersion from 0.5V to 0.8V is compatible with the 1% nominal tolerance of the resistors. The increase in uncertainties from ±0.04V to ±0.08V with increasing voltage reflects the cumulative effect of the tolerance of series resistors on voltage dividers—additional contributions include partial saturation of components near the upper limit of the range and increased thermal noise proportional to signal amplitude.

4.2 Current reading

Table 3 presents the results of the linearity tests of current sensors in the range of 1.52A to 5.20A. The values correspond to the averages of 10 successive measurements for each level, with uncertainties expanded between ±0.015A and ±0.020A. Graph 2 shows the linearity curves and the average error per channel.

Table 4. Current Sensor Validation Data.

Table 3- Current sensor validation data

I ref (A)	Channel 1 (A)	Channel 2 (A)	Channel 3 (A)	Channel 4 (A)	Channel 5 (A)	Channel 6 (A)
1,52	1,54 ±0,015	1,53 ±0,015	1,54 ±0,015	1,50 ±0,020	1,52 ±0,015	1,53 ±0,015
1,80	1,80 ±0,015	1,81 ±0,015	1,80 ±0,015	1,78 ±0,020	1,80 ±0,015	1,82 ±0,015
2,00	1,98 ±0,015	1,99 ±0,015	1,98 ±0,015	1,97 ±0,020	2,00 ±0,015	2,01 ±0,015
2,39	2,36 ±0,015	2,37 ±0,015	2,38 ±0,015	2,34 ±0,020	2,36 ±0,015	2,35 ±0,015
3,15	3,16 ±0,015	3,17 ±0,015	3,16 ±0,015	3,13 ±0,020	3,14 ±0,015	3,15 ±0,015
3,53	3,50 ±0,015	3,51 ±0,015	3,50 ±0,015	3,48 ±0,020	3,50 ±0,015	3,49 ±0,015
4,00	3,98 ±0,015	3,99 ±0,015	3,98 ±0,015	3,96 ±0,020	3,98 ±0,015	3,97 ±0,015
4,60	4,58 ±0,015	4,59 ±0,015	4,58 ±0,015	4,56 ±0,020	4,58 ±0,015	4,57 ±0,015
5,20	5,18 ±0,015	5,19 ±0,015	5,18 ±0,015	5,16 ±0,020	5,21 ±0,015	5,17 ±0,015

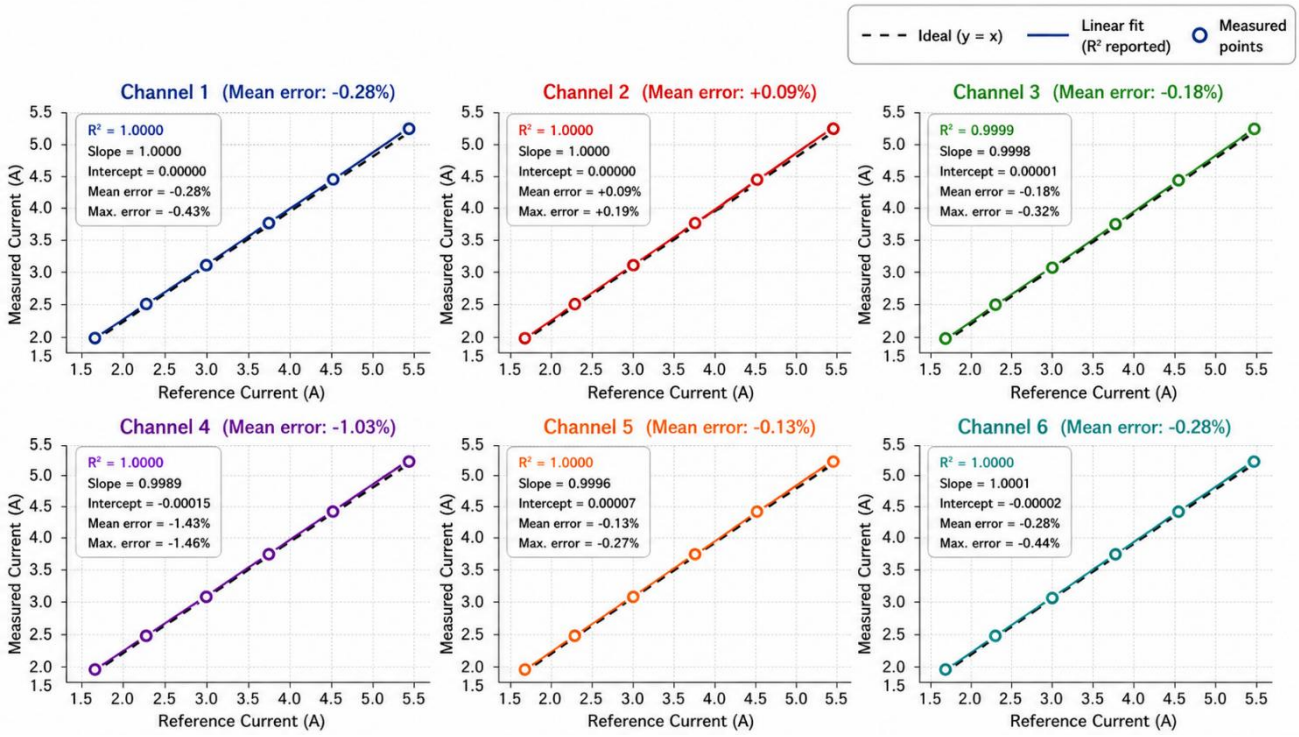

Figure 13. linearity of current sensors

Table 4- Current Sensor Performance Indicators

Channel	R^2	Average error (%)	Max. error (%)	Uncertainty (A)	ENL (%)
1	0,9999	−0,28	+1,32	±0,015	0,38
2	0,9999	+0,09	+1,11	±0,015	0,32
3	0,9999	−0,18	+1,32	±0,015	0,35
4	0,9998	−1,03	−1,32	±0,020	0,42
5	0,9999	−0,13	+1,11	±0,015	0,29
6	0,9999	−0,28	+0,50	±0,015	0,27

The linearity of the six current channels is satisfactory, with R^2 higher than 0.9998. The curves of all channels adhere to the ideal reference line, with practically overlapping behavior. Five channels have mean absolute errors of less than 0.01 A:

Channel 1 (−0.009 A, −0.28%), Channel 2 (+0.003 A, +0.09%), Channel 3 (−0.006 A, −0.18%), Channel 5 (−0.004 A, −0.13%), and Channel 6 (−0.009 A, −0.28%). Channel 4 presents a systematic underestimation of −0.033 A (−1.03%), suggesting variation in the transformation ratio or in the specific conditioning circuit of this channel. This deviation, although superior to the others, remains within acceptable limits. The inter-channel dispersion of 0.036 A represents approximately 0.98% of the tested measurement range.

The dispersion of repeated measurements has a distinct characteristic of voltage sensors: five channels exhibit constant uncertainty of ±0.015 A over the entire range, while Channel 4 exhibits ±0.020 A. The constancy of uncertainties in absolute amplitude, independent of the measured current, indicates that the variability is dominated by fixed-amplitude noise sources — thermal noise in the burden resistor, electronic noise from the conditioning and quantization circuit of the A/D converter — and not by the amplitude of the signal, different from the pattern observed in voltage sensors. Graph 3 presents a comparative view of all channels

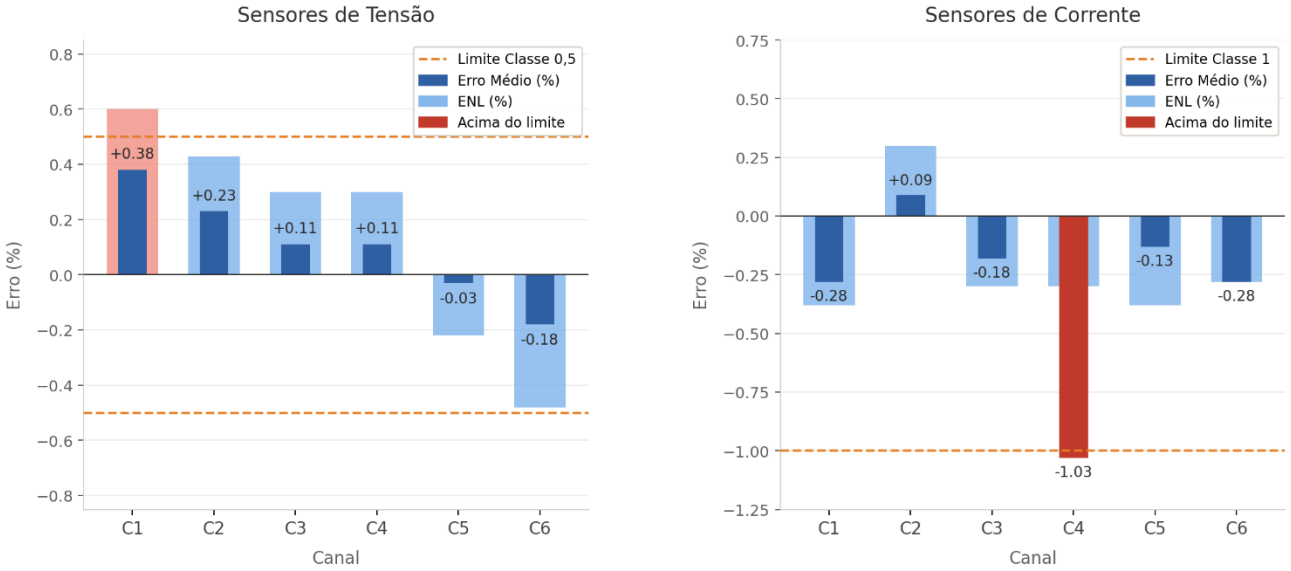

Figure 14. Summary of measurement errors by channel

Table 5-Overall Performance Indicators of the Measurement System

Quantity	Average R^2	Average error (%)	Max. error (%)	Inter-channel dispersion / Relative uncertainty
Voltage	0,99997	±0,17	±0,24	0,92 V (0,93%) / 0,04%
Current	0,99995	±0,30	±0,95	0,036 A (0,98%) / 0,5% (>3A)

4.3 Adequacy of the Measurement System

The measurement system presents adequate linearity for both quantities ($R^2 > 0.9998$), with mean errors of less than ±0.35% and inter-channel dispersion close to 1% of the measurement range. These results validate the proposed architecture based on IC ATM90E36 with resistive dividers for voltage and transformers ZMCT103C for current. Compared to Guevara et al. (2022), who reported errors of 0.9% for voltage and 7% for current with the M90E32AS chip, the present work achieves superior metrological performance — particularly for current, where the improvement is 23 times. This difference can be attributed to the use of ZMCT103C transformers and the systematic calibration process per individual channel. The linearity obtained and the analysis of inter-channel dispersion provide a solid foundation to feed the ML model with accurate data for anomaly detection.

4.4 ESP-NOW Communication

The integrity tests were conducted indoors, with a distance of 20 meters between transmitter and receiver, corresponding to the separation between classrooms with intermediate masonry walls. This condition represents a challenging scenario for wireless communication due to the attenuation by physical obstacles. Tables 6 and 7 and Figure 4 present the complete results.

Table 6-Success Rate in Data Transmission as a Function of the Interval Between Submissions

Interval (ms)	Packets sent	Packets received	Success rate (%)	Lost packets
50	1000	892	89,2	108
100	1000	967	96,7	33
250	1000	995	99,5	5
500	1000	999	99,9	1
1000	1000	1000	100,0	0

Table 7- Communication latency at different transmission intervals

Interval (ms)	Average latency (ms)	Minimum latency (ms)	Maximum latency (ms)	Standard Deviation (ms)
50	8,7	3,2	47,3	6,8
100	6,4	3,1	18,5	3,2
250	5,8	2,9	12,1	1,9
500	5,3	2,8	9,4	1,4
1000	4,9	2,7	8,2	1,1

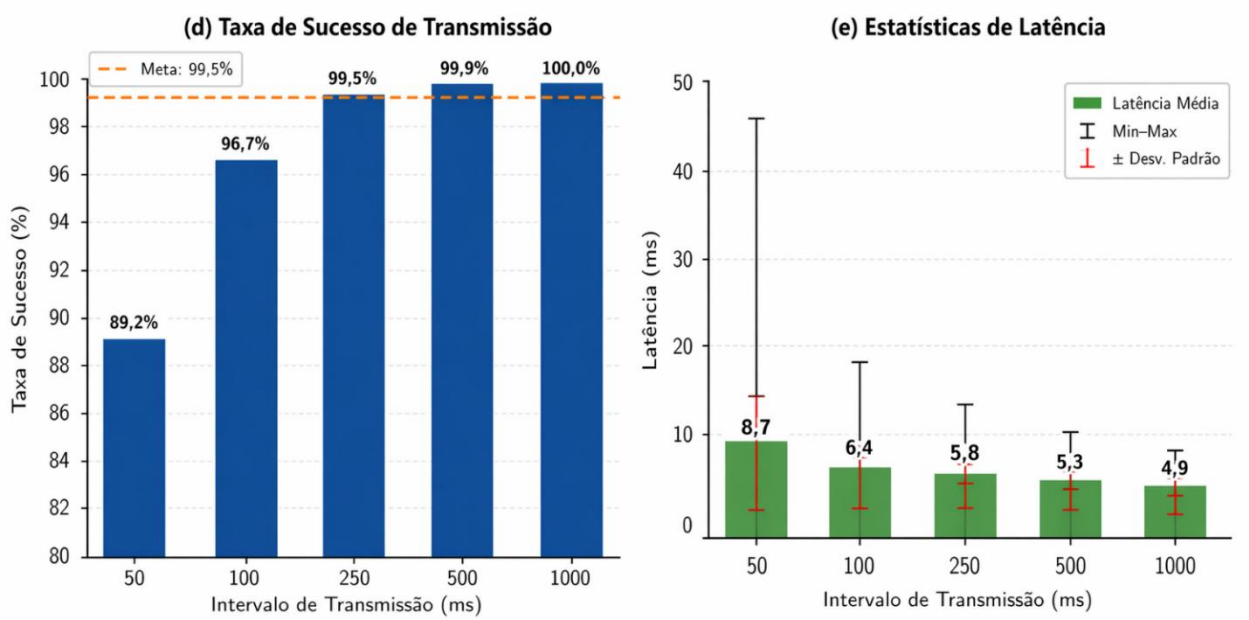

Figure 15. ESP-NOW success rate by transmission interval (left) and latency statistics with min–max error bars and ±standard deviation (dir).

Table 8- ESP-NOW Communication Performance Summary (reference range: 250 ms)

Metrics	Value (250 ms interval)	Suitability
Success rate	99,5%	Excellent
Loss Rate	0,5%	Within the acceptable range (<1%)
Average latency	5,8 ms	Suitable for real-time monitoring
Standard deviation of latency	1,9 ms	Low variability
Maximum latency	12,1 ms	Acceptable for the proposed application
Tested distance	20 m (with masonry walls)	Sufficient for building environments

The results demonstrate that the packet loss rate is directly related to the interval between consecutive transmissions. For 50 ms intervals, 10.8% of the packets were lost, indicating channel saturation or limitation in the processing of acknowledgments. The ESP-NOW protocol operates in half-duplex mode, where the transmission of acknowledgments competes with the reception of new packets, explaining the degradation observed at a high transmission rate. The analysis reveals exponential decreasing behavior of the loss rate for intervals of less than 100 ms.

For intervals of 250 ms or higher, the success rate exceeded 99.5%, reaching 100.0% with 1000 ms interval. This threshold represents the proper operating point of the system, where reliability is ensured without compromising latency. The mean latency was between 4.9 ms and 8.7 ms, with shorter intervals resulting in higher values and greater variability (standard deviation from 6.8 ms to 50 ms). For 250 ms or higher, latency has stabilized around 5 ms with standard deviation of less than 2 ms. Minimum latency of ~3 ms in all conditions represents the propagation and processing time under ideal conditions. Statistical analysis reveals that 95% of transmissions at 250 ms had latency lower than 9 ms.

For energy monitoring with a 1-second sampling rate, the module sends one packet per second—an interval 4 times higher than the minimum of 250 ms for 99.5% success. In the 1-minute setting, the interval is 240 times longer than the minimum required. The 5.8 ms latency represents less than 0.6% of the 1-second period, making it negligible for ML

model processing. The communication system is, therefore, fully adequate to support the proposed architecture of monitoring with centralized processing.

4.5 Machine Learning Model

Table 9 presents the performance comparison of the three models in the isolated test set of 216 samples. Graph 5 graphically illustrates the predictions compared to the actual test set value.

Table 9- Performance Comparison of ML Models in the Test Suite (216 samples)

Model	MAE (W)	RMSE (W)	MAPE (%)	R^2	Flash (%)
Baseline — Historical Average	91,53	112,27	14,85	-0,1743	68,6
Gradient Boosting Regressor	82,42	92,36	12,65	0,2051	75,0
Random Forest Regressor	70,71	108,96	10,91	0,2051	80,4

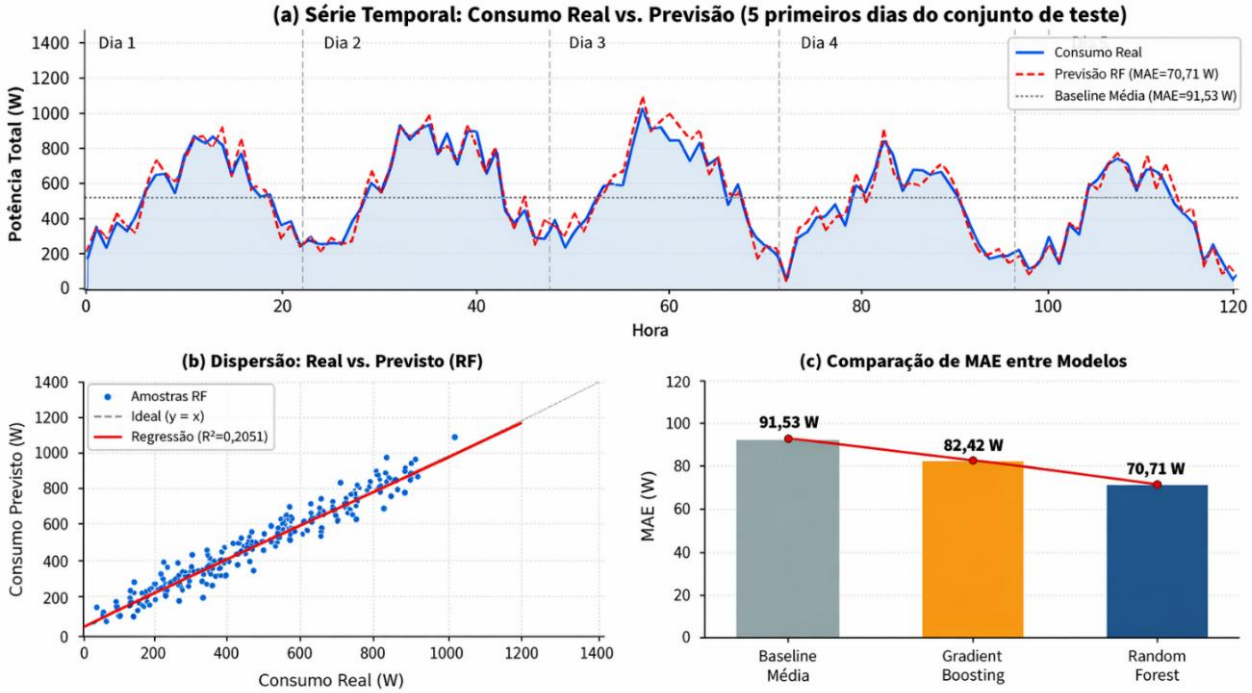

Figure 16. Comparative metrics: (a) actual vs. predicted time series in the first 5 days of the test; (b) actual vs. predicted dispersion (RF); (c) comparison of MAE between the three models

Table 10-- Performance of Models on the ESP32-S3 Microcontroller

Model	Flash Memory (%)	Inference time
Baseline — Historical Average	68,6%	< 5 ms
Gradient Boosting Regressor	75,0%	30 ms
Random Forest Regressor	80,4%	20 ms

The Random Forest Regressor model showed better overall performance, with MAE of 70.71 W (22.7% lower than baseline), MAPE of 10.91% and R^2 of 0.2051. Gradient Boosting had intermediate performance (MAE of 82.42 W), while baseline had MAE of 91.53 W. The graphs show that Random Forest is better able to capture consumption variations, while baseline maintains a constant forecast in the historical average. The MAE of 70 W over average consumption of ~720 W represents a relative error of 9.8% — a typical deviation operationally relevant for energy planning and demand response.

The performance analysis on the microcontroller (Table 10) demonstrates that the Random Forest Regressor simultaneously meets the two implementation criteria: (i) superior predictive performance — MAE of 70.71 W, 22.7% better than the baseline; and (ii) proven feasibility of embedded deployment, with 80.4% Flash memory usage and 20 ms inference. Gradient Boosting, despite lower Flash usage (75%), requires 30 ms of inference and has lower predictive performance. This convergence of criteria underlies the selection of Random Forest Regressor as the final model.

The R^2 of 0.2051 should be interpreted in the context of the characteristics of the dataset: Channel 3 presented a coefficient of variation of 80.9% (standard deviation of 363.1 W over a mean of 448.9 W), indicating loads with highly intermittent and stochastic behavior. Under these conditions, the model captures the systematic components of the signal—hourly and weekly patterns—while the residual variance corresponds to the genuinely unpredictable component of the loads. Because it is sensitive to the autocorrelation of time series, R^2 was not used

as an isolated selection criterion, as established in Section 3.6. The 70W MAE over ~720W average consumption, combined with 10.91% MAPE, is sufficient for trend-based management and proactive demand response.

5. COMPLETE SYSTEM PERFORMANCE

The integrated validation of the three subsystems confirms that the system operates consistently within the established requirements. Figures 17, 18, and 19 show the constructed boards: Figure 17 presents the central board, Figure 18 the 6-channel board, and Figure 19 the single-channel board. The metrological accuracy of the sensors ($R^2 > 0.9998$, average errors of ±0.17% for voltage and ±0.30% for current) ensures adequate quality of the input data for the ML model. ESP-NOW communication with a 99.5% success rate and a margin of 4× over the 1-second sampling period ensures transmission reliability. The embedded model performs inferences in 20 ms within a period of 1 second, enabling autonomous operation. This complete cycle—from edge measurement to local prediction to actuator control, with no reliance on router or cloud infrastructure—validates the technical feasibility of the proposed system.

Figure 17. Central board of the developed system

Figure 18. Six-channel board of the developed system.

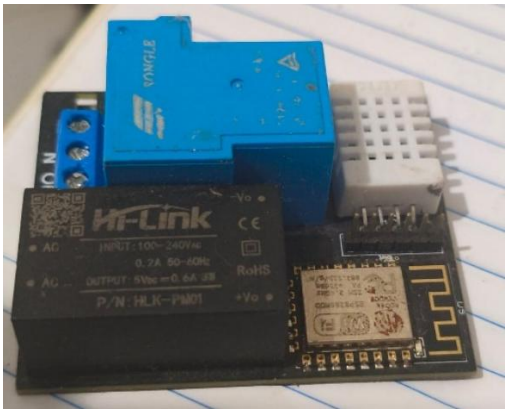

Figure 19. Single-channel board of the developed system.

6. CONCLUSION

This work presented the development and experimental validation of an energy management and optimization system in Buildings Based on Edge Computing, using the ESP32-S3 microcontroller and embedded Machine Learning. The three subsystems of the system were developed and experimentally validated with results that demonstrate the feasibility of the proposed approach.

The voltage and current sensors based on the ATM90E36 IC, individually calibrated per channel, achieved R^2 greater than 0.9998 with average errors of ±0.17% and ±0.30%, respectively, validating the architecture with resistive splitters and transformers ZMCT103C. The ESP-NOW communication demonstrated a success rate of 99.5% and an average latency of 5.8 ms at a 250 ms interval, at 20 meters with masonry obstacles, without the need for an external router. The embedded Random Forest Regressor model achieved MAE of 70.71 W with 20 ms inference on the ESP32-S3, confirming the feasibility of ML processing at the edge with autonomous operation.

The complete cycle — from edge measurement to local forecasting and load control, without dependence on external infrastructure — validates the proposal for autonomous building energy management. The solution is replicable in a variety of urban contexts and especially suitable for environments with limited telecommunications connectivity. As future works, the following stand out: (i) expansion of the collection to the six channels in a minimum period of 365 days; (ii) implementation and validation of the Isolation Forest (Autoencoder) anomaly detection module; (iii) improvement of temporal features with cyclic coding (sine/cosine) and lag features to improve the R^2 of the model; and (iv) validation in a larger operational building.

7. REFERENCES

[1] A. A. Sônego, R. Marcelino, and V. Gruber, "The Internet of Things applied to the concept of energy efficiency: a quantitative-qualitative analysis of the state of the art literature," *AtoZ: new practices in information and knowledge*, vol. 5, no. 2, p. 80, 2017, doi: 10.5380/atoz.v5i2.47860.

[2] N. E. Guevara, Y. H. Bolaños, J. P. Diago, and J. M. Segura, "HardwareX Development of a low-cost IoT system based on LoRaWAN for monitoring variables related to electrical energy consumption in low voltage networks," *HardwareX*, vol. 12, p. e00330, 2022, doi: 10.1016/j.ohx.2022.e00330.

[3] FCCC, "Paris Agreement," *Framework on Climate*

Change, vol. CP/2015/L. 2015.

[4] D. A. Verde Romero, E. Villalvazo Laureano, R. O. Jiménez Betancourt, and E. Navarro Álvarez, “An open source IoT edge-computing system for monitoring energy consumption in buildings,” *Results Eng.*, vol. 21, no. February, 2024, doi: 10.1016/j.rineng.2024.101875.

[5] J. dos S. Costa and L. M. L. de Andrade Junior, "Energy efficiency applied to electricity consumption: A literature review study," *Res. Soc. Dev.*, vol. 10, no. 4, p. e26210414085, 2021, doi: 10.33448/rsd-v10i4.14085.

[6] L. Da Xu, W. He, and S. Li, “Internet of things in industries: A survey,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014, doi: 10.1109/TII.2014.2300753.

[7] H. Hua, Y. Li, T. Wang, N. Dong, W. Li, and J. Cao, “Edge Computing with Artificial Intelligence: A Machine Learning Perspective,” *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–35, Sep. 2023, doi: 10.1145/3555802.

[8] H. J. El-Khozondar *et al.*, “A smart energy monitoring system using ESP32 microcontroller,” *e-Prime - Adv. Electr. Eng. Electron. Energy*, vol. 9, no. June, p. 100666, 2024, doi: 10.1016/j.prime.2024.100666.

[9] T. M. SERRANO, *DESIGN AND DEVELOPMENT OF AN IOT SOCKET WITH ADVANCED ELECTRIC ENERGY MEASUREMENT AND MONITORING*, vol. 2507, no. February. CAMPINAS, 2020.

[10] M. I. Joha, M. S. Nazim, and M. I. Zubair, “Smart Energy Metering, Control, and Protection System for Making a Smart Home Using IoT,” *2023 26th Int. Conf. Comput. Inf. Technol. ICCIT 2023*, no. December, pp. 1–6, 2023, doi: 10.1109/ICCIT60459.2023.10441201.

[11] U. Ali, M. U. Ramzan, W. Ali, M. E. Rana, and A. Qayyum, “IoT-Driven Smart Energy Monitoring: Real-time Insights and AI-Based Unit Predictions,” *2023 IEEE 21st Student Conf. Res. Dev. SCOReD 2023*, pp. 672–677, 2023, doi: 10.1109/SCOReD60679.2023.10563825.

[12] Francisco Borges Carreiro;, Júlia Ferreira Nascimento, and Guilherme Mateus Lopes Sousa, "Modeling and Development of an IoT and Digital Twin Solution for Residential Energy Consumption Measurement," *Procedings do XXIV Congr. Bras. Automatic*, pp. 1513–1520, 2022, doi: 10.20906/cba2022/3380.

[13] M. Khan, J. Seo, and D. Kim, “Towards energy efficient home automation: A deep learning approach,” *Sensors (Switzerland)*, vol. 20, no. 24, pp. 1–18, 2020, doi: 10.3390/s20247187.

[14] D. Alfonso, V. Romero, and E. V. Laureano, "An open-source IoT edge computing system for monitoring energy consumption in buildings," vol. 21, 2024.

[15] K. E. N. Peffers, T. Tuunanen, and M. A. Rothenberger, “A Design Science Research Methodology for Information Systems Research,” vol. 24, no. 3, pp. 45–77, 2008, doi: 10.2753/MIS0742-1222240302.

[16] A. Dresch, A. Proença, J. Antonio, and V. Antunes, "Design Science Research : research method for production engineering," pp. 741–761, 2013.

[17] M. BUSCHMANN, Frank; MEUNIER, regine; ROHNERT, Hans; SOMMERLAND, Peter; STAL, *SOFT WARE ARCHITECTURE*. New York, 1996.

[18] Hik, "DIGITAL MULTIMETER," vol. 189, no. L, pp. 1–4, 2000.

[19] MICROCHIP TECHNOLOGY IN, “Atmel M90E36A.” San Jose, 2016.

[20] S. Kapoor and A. Narayanan, “Article Leakage and the reproducibility crisis in machine- learning-based science Leakage and the reproducibility crisis in machine-learning-based science,” *Patterns*, vol. 4, no. 9, p. 100804, 2023, doi: 10.1016/j.patter.2023.100804.