

# AI-Enhanced Embedded IoT System for Real-Time Industrial Sensor Calibration <sup>†</sup>

Alan Cuenca-Sánchez <sup>1,\*</sup>, Jeampier Iza <sup>1</sup>, Pablo Proaño <sup>2</sup> and Javier Valenzuela <sup>1</sup>

<sup>1</sup> Escuela de Formación de Tecnólogos, Escuela Politécnica Nacional, Quito 170143, Ecuador; jeampier.izaalquinga@epn.edu.ec (J.I.); javier.valenzuela@epn.edu.ec (J.V.)

<sup>2</sup> Departamento de Automatización y Control Industrial, Escuela Politécnica Nacional, Quito 170143, Ecuador; pablo.proano@epn.edu.ec

\* Correspondence: alan.cuenca@epn.edu.ec

<sup>†</sup> Presented at the XXXIII Conference on Electrical and Electronic Engineering, Quito, Ecuador, 11–14 November 2025.

## Abstract

This study presents the design and validation of an AI-enhanced embedded IoT system for real-time industrial sensor calibration. The proposed platform integrates a PT100 temperature sensor and a 4–20 mA pressure transmitter with an ESP32 microcontroller, enabling on-device data acquisition, processing, and wireless transmission. A lightweight multilayer perceptron (MLP) neural network, trained in Python with a hybrid dataset (synthetic and experimental) and deployed on the ESP32 via JSON weight files, performs local inference to estimate ideal sensor outputs and compute key performance metrics. Experimental tests under controlled laboratory conditions confirmed high accuracy, with efficiency above 98.6%, RMSE below 0.005 V, and absolute uncertainty margins of  $\pm 0.5$  °C and  $\pm 0.07$  bar. Additionally, 95% confidence intervals for RMSE and standard deviation demonstrated statistical reliability across all operating points. The prototype also addresses practical constraints, including ESP32 ADC nonlinearity, energy consumption, and multi-sensor scalability, while remaining portable and low-cost. The integration of edge AI capabilities demonstrates the feasibility of executing accurate neural network models directly on embedded microcontrollers, eliminating reliance on cloud-based processing. The proposed solution provides a robust proof-of-concept that is scalable, cost effective, and suitable for industrial IoT applications, predictive maintenance, and Industry 4.0 environments, with future work focusing on long-term drift evaluation and validation under real industrial conditions.



Academic Editor: Jackeline Abad

Published: 15 November 2025

**Citation:** Cuenca-Sánchez, A.; Iza, J.; Proaño, P.; Valenzuela, J. AI-Enhanced Embedded IoT System for Real-Time Industrial Sensor Calibration. *Eng. Proc.* **2025**, *115*, 13. <https://doi.org/10.3390/engproc2025115013>

**Copyright:** © 2025 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/>).

**Keywords:** artificial intelligence; edge computing; embedded systems; ESP32 microcontroller; Industry 4.0; Internet of Things; neural networks; real-time monitoring; sensor calibration

## 1. Introduction

The reliability of industrial data acquisition systems strongly depends on sensor accuracy and proper calibration [1]. In critical applications, even small deviations in temperature or pressure can lead to operational failures, which motivates real-time calibration strategies [2]. Industrial transmitters also exhibit design-specific behaviors that must be considered during calibration and validation [3]. Recent works have proposed neural network approaches for error compensation in pressure sensing [4] and for improving temperature calibration [5], with related advances reported for other nonlinear sensors [6].

Embedded platforms provide a cost-effective path to integrate acquisition, preprocessing, and wireless communication in a single device [7]. Within this context, the ESP32 has become a popular option for industrial and cyber-physical applications due to its processing capabilities and integrated connectivity [8]. Its versatility is evidenced by deployments ranging from vehicle dynamics control [9] to the biomedical monitoring of vital signs [10]. Further use cases include asset tracking in logistics [11] and open-source SCADA architectures for energy systems [12]. Comparisons with alternative low-cost platforms indicate a favorable balance between performance and development effort [13]. In parallel, real-time field prototypes have addressed environmental monitoring [14] and hydraulic temperature supervision in heavy equipment [15], while recent studies discuss data acquisition pipelines tailored to ESP32-based systems [16]. Practical development is also supported by accessible tooling that facilitates rapid prototyping and debugging [17].

Artificial intelligence (AI), particularly artificial neural networks (ANNs), has been employed to compensate sensor nonlinearities and enable adaptive calibration policies on embedded targets [18]. Advances in edge AI now permit on-device inference, reducing latency and mitigating dependence on persistent connectivity [19]. From a systems perspective, executing inference locally at the edge can also improve predictability in time-critical loops [20]. On the modeling side, recent results clarify training and generalization aspects of lightweight networks suitable for deployment [21], and complementary probabilistic formulations inform error modeling and uncertainty handling [22]. Finally, emerging TinyML applications show the feasibility of predictive maintenance and fault detection directly on microcontrollers, reinforcing the relevance of embedded AI for industrial IoT [23].

This paper addresses these gaps by proposing an AI-enhanced embedded IoT system for real-time sensor calibration, deploying a multilayer perceptron (MLP) trained in Python and executed locally on the ESP32. The design explicitly considers constraints of embedded operation (ADC behavior, energy, and uncertainty propagation) and targets portability and scalability for industrial contexts. The main contributions of this work are as follows:

- The development of a low-cost embedded IoT platform integrating temperature and pressure sensors with an ESP32 for real-time calibration.
- The deployment of a lightweight MLP neural network directly on the microcontroller without cloud dependency.
- Experimental validation demonstrating efficiency above 95% with low RMSE and bounded uncertainty across tested ranges.
- Demonstration of a scalable and portable solution applicable to Industry 4.0 environments.

## 2. Methodology

The methodology followed a systematic approach to develop and validate the embedded IoT system. It integrates hardware, signal acquisition, data processing, and neural network inference, executed locally on the ESP32. The process comprised three stages: (i) system architecture design, (ii) neural network training and deployment, and (iii) on-device inference in real time.

### 2.1. System Architecture

The proposed data acquisition system was designed to integrate temperature and pressure measurements with real-time digital processing. Figure 1 shows the experimental prototype and the main components: the sensor module (including signal conditioning), the ESP32 microcontroller, the power supply, the pressurized air intake for testing the pressure transmitter, and the IoT interface for data visualization.

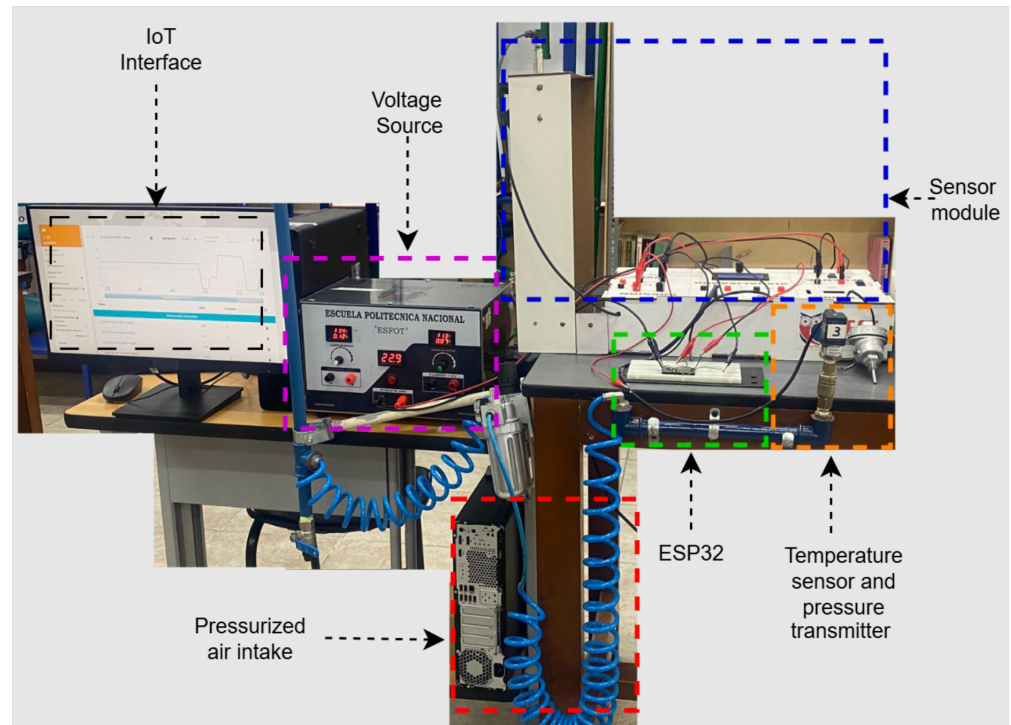


**Figure 1.** AI-enhanced data acquisition system showing the main components: IoT interface, power supply, pressurized air intake, ESP32 microcontroller, and sensor module with signal conditioning for the PT100 and pressure transmitter.

It is important to clarify that, since this work focuses on the development of a functional prototype, a dedicated printed circuit board (PCB) was not fabricated. All signal acquisition, preprocessing, and neural network inference are executed directly on the ESP32, which acts as the central processing unit. Therefore, the circuit implementation corresponds to direct connections of the PT100 and 4–20 mA pressure transmitter to the ESP32 ADC channels through the conditioning circuits included in the sensor module. This configuration ensured flexibility during testing while maintaining the low-cost and scalable nature of the platform.

## 2.2. Neural Network Integration

To enhance calibration accuracy, two lightweight multilayer perceptron (MLP) models were implemented: a temperature model trained to estimate ideal voltage from the measured temperature, and a pressure model trained to estimate ideal voltage from the measured pressure. Both models were trained in Python 3.10 using scikit-learn's MLPRegressor with one hidden layer (4 neurons, ReLU activation, linear output). The training dataset was built using a hybrid approach. Initial synthetic samples were generated from the calibration curves obtained in laboratory tests, expressed as linear Equations (1) and (2):

$$V(T) = 0.0366T + 0.008 \quad (1)$$

$$V(P) = 0.237P + 0.93 \quad (2)$$

where  $V(T)$  and  $V(P)$  represent the ideal voltages corresponding to a measured temperature  $T$  (°C) and pressure  $P$  (bar), respectively. The slope terms indicate the voltage change per unit of temperature or pressure, while the intercepts account for sensor and conditioning circuit offsets. In addition, experimental measurements collected during calibration sessions were incorporated into the training dataset. This hybrid strategy enabled the MLP to generalize beyond ideal linear responses and effectively correct nonlinear deviations

observed in practice. The trained models were then exported as JSON files for deployment on the ESP32, enabling local inference and real-time correction of sensor readings.

### 2.3. On-Device Inference and Real-Time Operation

The trained MLP models were deployed on the ESP32-WROOM-32 using MicroPython, selected for its flexibility in handling JSON files containing model weights and biases. During each acquisition cycle, the ESP32 performed (i) analog acquisition from ADC channels (GPIO34 for temperature, GPIO35 for pressure), (ii) preprocessing through moving average filtering and normalization, (iii) forward propagation with stored weights, biases, and activation functions (ReLU hidden layer, linear output), and (iv) computation of ideal voltage and performance metrics (efficiency, RMSE, standard deviation, and uncertainty).

Inference was executed in less than 5 ms, supporting a 1 Hz sampling rate without performance loss. Energy consumption was optimized by enabling Wi-Fi modem sleep between cycles. This stage ensured full local processing, avoiding cloud dependency while addressing ADC nonlinearity through MLP compensation. Figure 2 summarizes the acquisition, processing, and transmission workflow.

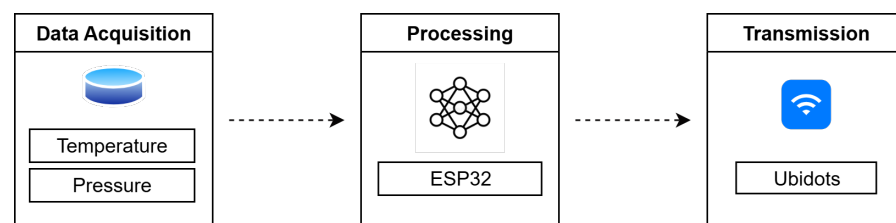


**Figure 2.** Workflow of acquisition, processing, and transmission for the on-device inference system implemented on the ESP32.

## 3. Implementation and Results

### 3.1. Experimental Configuration and Test Conditions

The ESP32 was connected to a PT100 sensor (GPIO34) and a 4–20 mA pressure transmitter (GPIO35), acquiring data at 1 Hz and processing it in real time with the deployed MLP models. For each test point, 200 repeated samples were collected to ensure repeatability and allow for the calculation of efficiency, RMSE, standard deviation, and uncertainty with statistical confidence.

Experiments were carried out under controlled laboratory conditions: (i) temperature range 20–90 °C using a thermal bath, (ii) pressure range 0–10 bar with an air compressor and precision regulator, and (iii) monitoring through the Thonny console (local values) and Ubidots dashboard (remote visualization).

The use of Ubidots was limited to data display, while all inference and calibration were executed locally on the ESP32. Figure 3 shows the laboratory prototype integrating the PT100 and pressure transmitter with the ESP32 through a conditioning module.

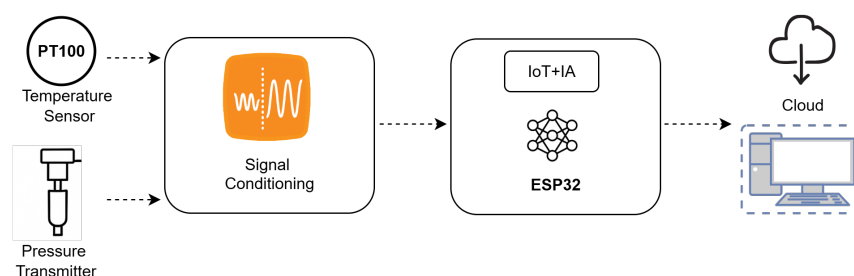


**Figure 3.** Laboratory setup for experimental testing of the embedded IoT system.

Figure 4 illustrates the cloud interface used for monitoring. This setup replicates industrial-like monitoring within a safe and reproducible laboratory environment.

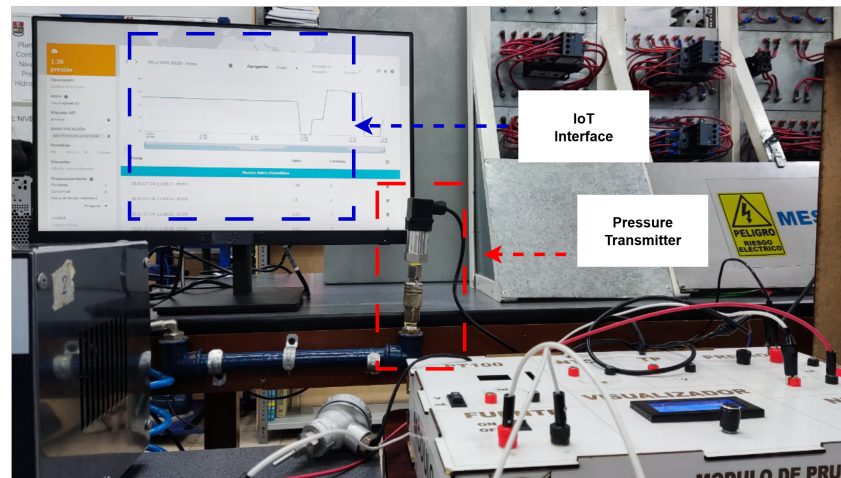


**Figure 4.** Remote visualization of real-time data on the Ubidots dashboard.

### 3.2. Temperature Calibration Performance

The PT100 sensor was tested from 20 to 90 °C with 200 repetitions per point under controlled conditions. Real-time predictions from the ESP32-MLP were compared with measured voltages, with representative results in Table 1. The MLP achieved efficiency >98.6%, RMSE <0.005 V, and deviation <0.021 V across the range, correcting minor errors above 75 °C. Uncertainty analysis gave  $\pm 0.5$  °C, with 95% CIs confirming result stability. Thus, the network compensates PT100 nonlinearities, enabling accurate real-time calibration.

**Table 1.** Real-time inference results for temperature.

| T (°C) | $V_{\text{measured}}$ (V) | $V_{\text{calculated}}$ (V) | Efficiency (%) | RMSE (V) | Std. Dev. (V) |
|--------|---------------------------|-----------------------------|----------------|----------|---------------|
| 90     | 3.28                      | 3.30                        | 99.3           | 0.0040   | 0.0125        |
| 78.6   | 2.91                      | 2.97                        | 99.1           | 0.0042   | 0.0132        |
| 50     | 1.84                      | 1.88                        | 98.9           | 0.0045   | 0.0150        |
| 26.5   | 0.98                      | 1.03                        | 98.8           | 0.0047   | 0.0180        |
| 20     | 0.74                      | 0.75                        | 98.6           | 0.0049   | 0.0205        |

Figure 5 confirms both the accuracy of the calibration and the suitability of the scaling for on-device acquisition in the ESP32-based monitoring system.

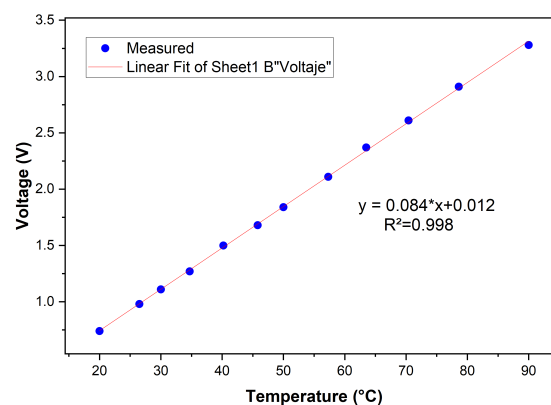


**Figure 5.** Calibration curve of the PT100 temperature sensor scaled to ESP32 ADC input range.

### 3.3. Pressure Calibration Performance

The pressure transmitter was evaluated from 0 to 10 bar in 2.5 bar increments under controlled laboratory conditions, with 200 repeated measurements per setpoint. The 4–20 mA signal was conditioned to the ESP32 ADC input range, and the MLP generated predicted values for calibration. Representative results are presented in Table 2.

**Table 2.** Real-time inference results for pressure.

| P (bar) | $V_{\text{measured}}$ (V) | $V_{\text{calculated}}$ (V) | Efficiency (%) | RMSE (V) | Std. Dev. (V) |
|---------|---------------------------|-----------------------------|----------------|----------|---------------|
| 10      | 3.30                      | 3.32                        | 99.4           | 0.0050   | 0.0110        |
| 7.5     | 2.92                      | 2.94                        | 99.3           | 0.0048   | 0.0125        |
| 5.0     | 2.50                      | 2.51                        | 99.2           | 0.0046   | 0.0132        |
| 2.5     | 2.05                      | 2.07                        | 99.1           | 0.0045   | 0.0145        |
| 0       | 0.93                      | 0.94                        | 99.0           | 0.0043   | 0.0155        |

Across the tested range, the MLP sustained efficiency above 99%, with RMSE below 0.005 V and standard deviation under 0.016 V. Small deviations at higher pressures, caused by transmitter nonlinearities, were corrected by the model. Uncertainty analysis yielded an absolute error of  $\pm 0.07$  bar, while 95% confidence intervals confirmed the stability of RMSE and deviation across all setpoints. Figure 6 shows the calibration curve in the 0–10 bar range, confirming both calibration accuracy and the suitability of the conditioning circuit for ESP32-based acquisition.

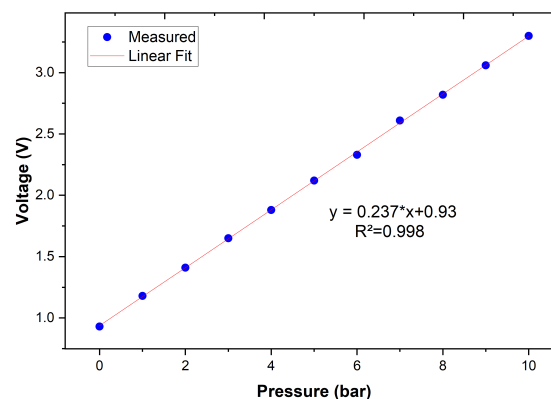


**Figure 6.** Calibration curve of the pressure transmitter scaled to the ESP32 ADC input range.

### 3.4. Discussion and System Robustness

The proposed platform embodies a low-cost design primarily through the use of the ESP32 microcontroller, a device integrating dual-core processing, wireless connectivity, and multiple ADC channels in a single unit [24,25]. This eliminates the need for external gateways or high-performance servers, thereby reducing both cost and complexity. The integration of temperature and pressure sensors with direct connections to the ESP32 ensures that acquisition, processing, and calibration can be performed entirely on-board, making the system portable and scalable.

The experimental results confirmed the capability of the system to perform accurate real-time calibration. Efficiency remained consistently above 98.6% for temperature and 99% for pressure, with RMSE values below 0.005 V for both variables. Low standard deviation values indicated stable signal behavior with minimal noise and drift. Performance was maintained within the tested ranges (20–90 °C and 0–10 bar). Although these intervals do not cover the full nominal ranges of the sensors (−50 to 200 °C for PT100 and 0–25 bar for the pressure transmitter), they were selected based on equipment availability and safety

considerations. Validation at the extreme operating points will be conducted in future work. Regarding the neural network architecture, a single hidden-layer MLP with four neurons was deliberately chosen as a trade-off between accuracy and computational efficiency. Other works explored higher-complexity models or comparative strategies [26–28], but their computational cost would compromise real-time execution on the ESP32. The selected MLP provided a practical baseline that balances performance with hardware limitations. Future work will extend this evaluation by comparing alternative architectures such as CNNs or polynomial regressions. A detailed uncertainty analysis was also performed. Contributions from the PT100 sensor ( $\pm 0.3\text{ }^{\circ}\text{C}$ ), the pressure transmitter ( $\pm 0.05\text{ bar}$ ), the ESP32 ADC resolution ( $\pm 0.01\text{ V}$ ), and the neural network residual error ( $< 0.005\text{ V}$ ) were combined, resulting in overall uncertainty margins of approximately  $\pm 0.5\text{ }^{\circ}\text{C}$  and  $\pm 0.07\text{ bar}$  within the tested ranges. In addition, 95% confidence intervals were calculated for RMSE and standard deviation values using repeated sampling (200 measurements per setpoint). RMSE remained below  $0.005\text{ V}$  and standard deviation below  $0.021\text{ V}$ , with narrow intervals, confirming the statistical reliability and robustness of the calibration results. Similar works in acquisition and filtering [29] have not reported such confidence analysis, highlighting the added methodological rigor of this study. Certain limitations must also be highlighted. The nonlinearity of the ESP32 ADC may introduce residual measurement errors; in this work, such effects were partially compensated by the MLP, although explicit ADC calibration was not performed. Power consumption tests indicated an average of  $120\text{ mA}$  during acquisition and processing, reduced to  $15\text{ mA}$  in modem-sleep mode, consistent with prior reports of energy-aware IoT deployments [30]. This level of consumption is suitable for short-term battery-powered use, but long-term drift phenomena and sensor aging remain open challenges. Likewise, although the prototype validated two sensors, simulations suggest that up to 6–8 channels could be managed on a single ESP32 through multiplexing and optimized memory allocation.

All experiments were conducted under controlled laboratory conditions, ensuring reproducibility but not fully capturing industrial factors such as vibration, EMI, or temperature fluctuations. Wireless protocols like ESP-NOW [31] and synchronized sensor networks [25] could improve robustness in such environments. Future field tests will confirm scalability and resilience. Overall, the system is suitable for predictive maintenance, Industry 4.0 with low-cost AI-enabled nodes, and educational or laboratory use as a portable calibration platform.

#### 4. Conclusions

This work presented the design and validation of an AI-enhanced embedded IoT system for real-time industrial sensor calibration, fully executed on a low-cost ESP32 platform. The system successfully integrated a PT100 temperature sensor and a 4–20 mA pressure transmitter, demonstrating high accuracy with efficiency above 98.6% for temperature and 99% for pressure. The inclusion of a lightweight multilayer perceptron (MLP) enabled local inference without dependency on external servers, reducing latency and infrastructure costs. We recognize that this study represents an initial laboratory proof-of-concept, and that certain methodological aspects remain to be extended. The tests were limited to controlled conditions and partial sensor ranges ( $20\text{--}90\text{ }^{\circ}\text{C}$  and  $0\text{--}10\text{ bar}$ ). Long-term drift, robustness against vibrations or electromagnetic interference, and validation under industrial operating environments are topics for future research. The revised analysis strengthened the methodological rigor of this study. Section 3.4 incorporated a detailed uncertainty evaluation, combining sensor tolerance, ADC resolution, and neural network residuals, resulting in overall margins of  $\pm 0.5\text{ }^{\circ}\text{C}$  and  $\pm 0.07\text{ bar}$ . Furthermore, 95% confidence intervals for RMSE and standard deviation confirmed the stability and statistical reliability of the

results. Additional discussion addressed ESP32 ADC nonlinearity, power consumption (120 mA in active mode, 15 mA in sleep mode), and scalability up to 6–8 channels, clarifying the capabilities and limitations of the proposed platform. In conclusion, the system demonstrates that accurate sensor calibration can be achieved directly on a microcontroller without reliance on cloud computing, offering a portable, low-cost, and scalable solution. With further development and industrial validation, the proposed platform has the potential to evolve into a benchmark for low-cost edge AI in real-time industrial sensor calibration, contributing to Industry 4.0 and predictive maintenance applications.

**Author Contributions:** Conceptualization, J.V.; Methodology, J.I.; Software, A.C.-S.; Validation P.P. and A.C.-S.; Formal Analysis, P.P.; Investigation, J.V. and J.I.; Writing—Original Draft Preparation, A.C.-S.; Writing—Review and Editing, P.P.; Visualization, J.V.; Supervision, J.I. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data are contained within the article.

**Acknowledgments:** During manuscript preparation, ChatGPT (OpenAI, GPT-4, 2025) was used to improve the English translation and readability. All AI-assisted content was reviewed and approved by the authors, who take full responsibility for the final version. No generative AI was used to create original ideas, text, or analyses.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Fraden, J. *Handbook of Modern Sensors: Physics, Designs, and Applications*, 5th ed.; Springer: Cham, Switzerland, 2016. [\[CrossRef\]](#)
2. Wang, J.; Zhang, H. Investigate Calibration Methods for Pressure Transmitters. *J. Phys. Conf. Ser.* **2020**, *1646*, 012084. [\[CrossRef\]](#)
3. Nadezhdin, I.; Goryunov, A. Differential Pressure Transmitter with Unified Electronics Unit. *IEEE Sens. J.* **2020**, *20*, 10460–10468. [\[CrossRef\]](#)
4. Almassri, A.; Hasan, W.; Ahmad, S.; Shafie, S.; Wada, C.; Horio, K. Self-Calibration Algorithm for a Pressure Sensor with a Real-Time Approach Based on an Artificial Neural Network. *Sensors* **2018**, *18*, 2561. [\[CrossRef\]](#)
5. Liu, C.; Zhao, C.; Wang, Y.; Wang, H. Machine-Learning-Based Calibration of Temperature Sensors. *Sensors* **2023**, *23*, 7347. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Tian, D.; Li, Y.; Jin, M.; Guo, Y.; Meng, F.; Liu, Q. Data-Driven Calibration of Tri-Axial Magnetic Sensors. *Meas. Sci. Technol.* **2025**, *36*, 035102. [\[CrossRef\]](#)
7. Samal, L.; Bute, P. Wireless Network for Industrial Application Using ESP32 as Gateway. In Proceedings of the 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 6–8 July 2023; pp. 1–5. [\[CrossRef\]](#)
8. Carducci, C.; Monti, A.; Schraven, M.; Schumacher, M.; Mueller, D. Enabling ESP32-Based IoT Applications in Building Automation Systems. In Proceedings of the 2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&IoT), Naples, Italy, 4–6 June 2019; pp. 306–311. [\[CrossRef\]](#)
9. Guzman, A.; Lua, C.; Garcia-Rodriguez, J.; Vidrios-Serrano, C.; Meza-Aguilar, M. Real-Time Embedded Control of Vehicle Dynamics Using ESP32: A Discrete Nonlinear Approach. *Electronics* **2024**, *13*, 3967. [\[CrossRef\]](#)
10. Rahman, M.; Li, Y.; Nabeed, T.; Rahman, M. Remote Monitoring of Heart Rate and ECG Signal Using ESP32. In Proceedings of the 2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Changsha, China, 26–28 March 2021; pp. 604–610. [\[CrossRef\]](#)
11. Goyal, B.; Dixit, K.; Dogra, A.; Nagar, M.; Akram, S.; Kaur, J. Empowering Assets and Vehicles with Cutting-Edge ESP32 Real-Time Tracking System. In Proceedings of the 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 28 February–1 March 2024; pp. 504–509. [\[CrossRef\]](#)
12. He, W.; Baig, M.; Iqbal, M. An Open-Source Supervisory Control and Data Acquisition Architecture for Photovoltaic System Monitoring Using ESP32, Banana Pi M4, and Node-RED. *Energies* **2024**, *17*, 2295. [\[CrossRef\]](#)

13. Jaafar, H.; Aminuddin, N. Comparison between NodeMCU ESP8266 and Uno R3 for Software Development Using Ubidots. *Malays. J. Bioeng. Technol.* **2024**, *1*, 36–40. [\[CrossRef\]](#)
14. Zafar, S.; Miraj, G.; Baloch, R.; Murtaza, D.; Arshad, K. An IoT Based Real-Time Environmental Monitoring System Using Arduino and Cloud Service. *Eng. Technol. Appl. Sci. Res.* **2018**, *8*, 3238–3242. [\[CrossRef\]](#)
15. Linggarjati, J. Design and Prototyping of Temperature Monitoring System for Hydraulic Cylinder in Heavy Equipment Using ESP32 with Data Logging and WiFi Connectivity. *IOP Conf. Ser. Earth Environ. Sci.* **2022**, *998*, 012042. [\[CrossRef\]](#)
16. Tran, V.; Thai, B.; Pham, H.; Nguyen, V.; Nguyen, V. A Proposed Approach to Utilizing ESP32 Microcontroller for Data Acquisition. *J. Eng. Technol. Sci.* **2024**, *56*, 474–488. [\[CrossRef\]](#)
17. Annamaa, A. Introducing Thonny, a Python IDE for Learning Programming. In Proceedings of the 15th Koli Calling Conference on Computing Education Research, Koli, Finland, 19–22 November 2015. [\[CrossRef\]](#)
18. Xu, X.; Ding, Y.; Hu, S.; Niemier, M.; Cong, J.; Hu, Y.; Shi, Y. Scaling for Edge Inference of Deep Neural Networks. *Nat. Electron.* **2018**, *1*, 216–222. [\[CrossRef\]](#)
19. Sakr, F.; Bellotti, F.; Berta, R.; De Gloria, A. Machine Learning on Mainstream Microcontrollers. *Sensors* **2020**, *20*, 2638. [\[CrossRef\]](#)
20. Li, E.; Zeng, L.; Zhou, Z.; Chen, X. Edge AI: On-Demand Accelerating Deep Neural Network Inference via Edge Computing. *IEEE Trans. Wirel. Commun.* **2019**, *19*, 447–457. [\[CrossRef\]](#)
21. Chen, S.; Dou, Z.; Goel, S.; Klivans, A.; Meka, R. Learning Narrow One-Hidden-Layer ReLU Networks. *arXiv* **2023**, arXiv:2304.10524. [\[CrossRef\]](#)
22. Raissi, M.; Perdikaris, P.; Karniadakis, G. Machine Learning of Linear Differential Equations Using Gaussian Processes. *J. Comput. Phys.* **2017**, *348*, 683–693. [\[CrossRef\]](#)
23. Gupta, S.; Shivhare, S. Embedded TinyML for Predictive Maintenance: Vibration Analysis on ESP32 with Real-Time Fault Detection in Industrial Equipment. *Int. J. Comput. Model. Appl.* **2025**, *2*, 1–17. [\[CrossRef\]](#)
24. Babiuch, M.; Foltyniek, P.; Smutny, P. Using the ESP32 Microcontroller for Data Processing. In Proceedings of the 2019 20th International Carpathian Control Conference (ICCC), Krakow-Wieliczka, Poland, 26–29 May 2019; pp. 1–6. [\[CrossRef\]](#)
25. Espinosa-Gavira, M.; Aguera-Perez, A.; Palomares-Salas, J.; Sierra-Fernandez, J.; Remigio-Carmona, P.; De-La-Rosa, J. Characterization and Performance Evaluation of ESP32 for Real-Time Synchronized Sensor Networks. *Procedia Comput. Sci.* **2024**, *237*, 261–268. [\[CrossRef\]](#)
26. Dokic, K.; Radisic, B.; Cobovic, M. MicroPython or Arduino C for ESP32—Efficiency for Neural Network Edge Devices. In *Intelligent Computing Systems. ISICS 2020; Communications in Computer and Information Science*; Springer: Cham, Switzerland, 2020; pp. 33–43. [\[CrossRef\]](#)
27. Dokic, K.; Martinovic, M.; Radisic, B. Neural Networks with ESP32—Are Two Heads Faster than One? In Proceedings of the 2020 6th Conference on Data Science and Machine Learning Applications (CDMA), Riyadh, Saudi Arabia, 4–5 March 2020; pp. 141–145. [\[CrossRef\]](#)
28. Chua, K.H.; Abu, M.A.; Shapiai, M.I.; Haniff, M.F.; Mohamad, R.S.; Abu, A. Analysis of Wind Speed Prediction using Artificial Neural Network and Multiple Linear Regression Model using Tynym on Esp32. *J. Adv. Res. Fluid Mech. Therm. Sci.* **2023**, *107*, 29–44. [\[CrossRef\]](#)
29. Mao, W.; Zhu, R.; Lu, Y.; Zhang, Z. Design of Adaptive Filter for Weak Signal Acquisition System. *J. Phys. Conf. Ser.* **2021**, *2026*, 012032. [\[CrossRef\]](#)
30. Hernandez, S.; Bulut, E. WiFi Sensing on the Edge: Signal Processing Techniques and Challenges for Real-World Systems. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 46–76. [\[CrossRef\]](#)
31. Mutiara, H.A.K.; Suhendra, T. Field Testing and QoS Analysis of ESP-NOW Communication on ESP32. In Proceedings of the 2024 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE), Badung, Indonesia, 24–25 October 2024; pp. 82–88. [\[CrossRef\]](#)

**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.