machines

MDPI

Article

Wavelet-Based Computational Intelligence for Real-Time Anomaly Detection and Fault Isolation in Embedded Systems †

Jesus Pacheco *, Victor H. Benitez, Guillermo Pérez and Agustín Brau

Departamento de Ingeniería Industrial, Campus Hermosillo, Universidad de Sonora, Hermosillo 83000, Mexico; victor.benitez@unison.mx (V.H.B.); a222230081@unison.mx (G.P.); agustin.brau@unison.mx (A.B.)

* Correspondence: jesus.pacheco@unison.mx

† This article is the expanded conference version of Perez, G.; Pacheco, J.; Benitez, V. Anomaly Behavior Analysis for Sensors Fault Detection. In Proceedings of the 2023 IEEE Symposium 479 Series on Computational Intelligence, SSCI 2023; pp. 1718–1723.

Abstract: In today's technologically advanced landscape, sensors feed critical data for accurate decision-making and actions. Ensuring the integrity and reliability of sensor data is paramount to system performance and security. This paper introduces an innovative approach utilizing discrete wavelet transforms (DWT) embedded within microcontrollers to scrutinize sensor data meticulously. Our methodology aims to detect and isolate malfunctions, misuse, or any anomalies before they permeate the system, potentially causing widespread disruption. By leveraging the power of wavelet-based analysis, we embed computational intelligence directly into the microcontrollers, enabling them to monitor and validate their outputs in real-time. This proactive anomaly detection framework is designed to distinguish between normal and aberrant sensor behaviors, thereby safeguarding the system from erroneous data propagation. Our approach significantly enhances the reliability of embedded systems, providing a robust defense against false data injection attacks and contributing to overall cybersecurity.

Keywords: discrete wavelet transform; embedded systems; anomaly behavior analysis; sensor fault detection; computational intelligence; cybersecurity; false data injection; anomaly detection

Citation: Pacheco, J.; Benitez, V.H.; Pérez, G.; Brau, A. Wavelet-Based Computational Intelligence for Real-Time Anomaly Detection and Fault Isolation in Embedded Systems. *Machines* **2024**, *12*, 664. https://doi.org/10.3390/machines12090664

Academic Editor: Alma Y. Alanis

Received: 22 August 2024
Revised: 17 September 2024
Accepted: 19 September 2024
Published: 22 September 2024

1. Introduction

In the rapidly evolving technological landscape, sensors are integral components that provide critical data for decision-making processes across various applications. Ensuring the reliability and accuracy of sensor data is critical, as any discrepancies can lead to significant errors and system failures. The importance of this issue is magnified in the context of embedded systems and the Internet of Things (IoT), where sensors operate in diverse and often harsh environments [1–7].

The challenge of ensuring sensor data integrity is amplified in embedded systems and IoT networks, where sensors are often exposed to harsh and variable conditions. Traditional methodologies for anomaly detection, such as signature-based Intrusion Detection Systems (IDS), have proven effective against known threats but struggle with novel and sophisticated attacks [8]. As cyber threats and system vulnerabilities evolve, there is a pressing need for advanced detection techniques that can adapt to emerging risks and safeguard system integrity.

This research addresses this critical gap by proposing an innovative approach that leverages discrete wavelet transforms (DWT) embedded within microcontrollers for real-time anomaly detection and fault isolation. Wavelet transforms, particularly DWT, have emerged as powerful tools in signal processing. They allow signals to be decomposed into different frequency components and analyze localized features. The Haar wavelet from the DWT family was chosen for anomaly detection due to its simplicity and efficiency in

decomposing non-stationary signals, allowing the detection of both transient and persistent faults in sensor data. Haar wavelets provide clear time and frequency localization, crucial for identifying anomalies in embedded systems. Their computational simplicity makes them ideal for real-time applications on resource-constrained devices like microcontrollers [9–11]. Euclidean distance was used together with DWT to quantify deviations between transformed data and a reference model, offering a straightforward and efficient way to detect faults.

The motivation behind this study lies in the recognition that traditional methods are insufficient for modern, dynamic environments where sensor data must be scrutinized continuously and accurately. By embedding wavelet-based analysis directly into microcontrollers, this approach allows for the meticulous monitoring of sensor data, enabling the detection and isolation of anomalies before they can propagate and disrupt the system.

The proposed scheme enhances system reliability and security by incorporating wavelet-based analysis into microcontrollers, which offers a robust defense against false data injection and other anomalies. This proactive approach not only improves the reliability of embedded systems but also contributes to overall cybersecurity by preventing erroneous data from compromising system integrity.

This paper leans on the work introduced in [12], targeting anomaly behavior analysis of sensors, the key differences are an innovative methodology for sensor fault detection using wavelet transforms and Euclidean distance calculations in embedded systems. We validate our approach through experiments involving embedded systems that simulate IoT network nodes. The results demonstrate a high detection rate with minimal false alarms, underscoring the efficacy of wavelet-based anomaly detection in enhancing sensor reliability and overall system security.

1.1. Background

With the raising use of IoT applications such as command and control, monitoring, and premises management, to name a few, ensuring the correct operation of all interconnected devices is becoming more and more challenging [8]. Predominantly, the proper functioning of sensors is crucial in IoT applications, requiring continuous research to detect and resolve device issues. Additionally, protecting data integrity as well as reducing potential risks are significant concerns for regular customers, businesses, and even researchers.

Recent research emphasizes the growing importance of wavelet-based methods for anomaly detection in embedded systems, particularly due to their ability to address the limitations of traditional fault detection approaches. For instance, the discrete wavelet neural network algorithm has demonstrated significant promise in detecting faults in rotor systems by effectively capturing signal features across multiple scales, resulting in enhanced detection accuracy and robustness [13]. Similarly, wavelet-based multi-class support vector machines have proven effective in diagnosing stator faults in induction motors, showcasing the method's capability to handle complex, multi-class fault scenarios with high precision [14]. Additionally, ensemble convolution-based methods for fault detection using vibration signals have shown that integrating multiple analytical approaches, including wavelet transforms, can significantly improve fault detection performance by leveraging diverse signal characteristics [15]. These studies highlight the advantages of wavelet-based techniques in decomposing signals into localized frequency components, making them well-suited for detecting both transient and persistent anomalies in real-time applications. By embedding wavelet analysis, fault detection can be substantially enhanced, contributing to the overall reliability and security of critical applications in dynamic environments. This evidence supports the foundation for the selection of wavelet-based methods in the current work, pointing out their potential to overcome the challenges posed by traditional anomaly detection techniques.

The next sections show a summary of the most frequent types of sensor faults, some of the strategies applied to address these issues, and the current security techniques for data integrity.

1.1.1. Sensor Fault Classification

The classification and detection of sensor faults have gathered significant attention due to the critical role of sensors in modern automated and industrial systems. Sensor faults are generally categorized into incipient and abrupt types, with incipient faults representing gradual deviations that can evolve into more severe issues over time, while abrupt faults occur suddenly and can have immediate detrimental effects on system performance [1]. Recent advances in fault classification methodologies reflect a growing trend towards leveraging sophisticated computational techniques. Deep learning approaches, such as convolutional autoencoders, have shown considerable potential in capturing complex fault patterns through unsupervised feature learning [2,3]. Additionally, machine learning algorithms like decision trees and particle swarm optimization have been applied to fault classification, providing interpretable models with adaptive capabilities [4]. More advanced methods, including support vector machines (SVM), K-nearest neighbor (KNN) classifiers, and generative adversarial networks (GAN), combined with random forests, have further enhanced fault classification accuracy by exploiting high-dimensional data characteristics and reducing false positives [5,6]. Time-frequency analysis (TFA) integrated with deep learning techniques has been successfully employed for the detection and classification of faults in complex systems such as unmanned aerial vehicles (UAVs), demonstrating the applicability of these methods in dynamic environments [7]. In highly sensitive settings like nuclear power plants, two-layer mathematical models utilizing data-driven methods have been developed for thermocouple sensor fault detection, showcasing the integration of statistical and machine learning approaches in critical safety applications [16]. These diverse techniques underline the ongoing global efforts and technological advancements aimed at improving sensor fault detection and classification, reflecting the growing complexity and critical importance of maintaining sensor integrity in varied applications across industries.

Loss of accuracy is the result of sensor bias, where data samples are replaced with constant values. Since it is a common issue, many strategies have been applied, for instance, authors in [17] suggest the use of angular-rate-aided estimation methods to improve bias assessment. Authors in [18] studied the bias issue in accelerometers used in the drilling process and introduced a sensor fault detection and isolation method that outperforms previous strategies.

The drift phenomenon in sensors is the presence of an offset or bias parameter that slowly changes (drifts) over time. Authors in [19] introduced a sensor drift detection method based on grey models and discrete wavelet transform (DWT), using DWT to decompose the signal and grey models for detrending. On the other hand, authors in [20] proposed a trinomial distribution for quantifying sensor drift in temperature sensors; the core of the method uses probabilistic neural networks (PNN) to estimate the correct temperature and then compare with the online values.

In addition to the drift and bias errors, a sudden fault can occur when the sensor unexpectedly stops working due to physical damage. This leads to a detectable fault parameter [1] that can be seen as sensor noise, short circuits, open circuits and/or random sensor faults. Sensor noise can show up in two types: internally (originating from the sensor and its inner circuit) and externally (occurring from an outside source). For instance, faulty connections and disconnections, respectively, trigger short and open-circuit faults. Random sensor faults, on the other hand, stem from the intricate layout environment, potentially surpassing the sensor's capabilities [1].

1.1.2. Discrete Wavelet Transform (DWT)

Wavelets are short-duration waveforms used for analyzing functions. Wavelets allow splitting signals into different frequency components and at different scales. Wavelet analysis involves decomposing signals into wavelet coefficients, representing their components at different scales and positions in time. This is different from approaches like the Fourier transform [9,10]. Due to these features, wavelets are widely used for solving problems related to time-varying non-stationary variables. The wavelet transform represents a signal

as a set of essential functions (wavelets) obtained from the translation and scaling of a mother wavelet, given by Equation (1):

$$\psi^*_{a,b} = \frac{1}{\sqrt{a}}\,\psi\left(\frac{t-b}{a}\right), a > 0, -\infty < b < \infty, \tag{1}$$

where $\psi^*_{a,b}$ represents the wavelet function, the core function used in the wavelet transform. a is the scaling parameter that stretches or compresses the wavelet, allowing analysis at different scales, and must be greater than zero. b is the translation parameter that shifts the wavelet along the time axis, allowing the function to be localized in time. t is the time variable representing the original signal's domain. $\psi\left(\frac{t-b}{a}\right)$ represents the mother wavelet function, which is the prototype for generating all the wavelets used in the analysis.

Continuous wavelet transforms (CWT) enable the mapping of properties in non-stationary signals. In time-frequency, the coefficients $W_f(a,b)$ in (2) are obtained by changing the scale and position parameters of a signal [13]:

$$W_f(a,b) = \int_{-\infty}^{\infty} f(t)\psi^*_{a,b}(t)dt \tag{2}$$

where $W_f(a,b)$ represents the wavelet coefficients, which quantify the similarity between the signal and the scaled and shifted wavelet function at a given scale and position. These coefficients are the results of the wavelet transform and describe the signal's behavior in both time and frequency domains. $f(t)$ is the original function being analyzed. It represents the time-domain data to which the wavelet transform is applied. $\psi^*_{a,b}(t)$ is the conjugate of the wavelet function, where the wavelet is scaled by a and translated by b. This function is used to match and extract specific features of the signal at different scales and positions.

There is also a discrete wavelet transform (DWT), which consists of the decomposition of the signal into a mutually orthogonal set of wavelets. Unlike traditional methods that analyze signals in the frequency domain only, DWT provides both time and frequency localization, making it effective for analyzing non-stationary signals commonly encountered in embedded systems [9]. In the context of sensor fault detection, DWT enables the isolation of specific features associated with faults, such as abrupt changes or gradual drifts, by decomposing the signal into various scales and resolutions. This multi-resolution analysis allows for the precise identification of anomalies that might be overlooked by other signal processing techniques. Additionally, the computational efficiency of DWT makes it suitable for real-time applications on resource-constrained embedded systems, where timely and accurate anomaly detection is crucial. The ability to perform detailed analysis at different levels of decomposition provides a robust framework for fault detection, aligning with the critical need to safeguard sensor data integrity and ensure reliable operation in complex and dynamic environments.

The DWT is described thoroughly in [11] and in the seminal paper [21]. This transform is expressed by (3)

$$\psi_{j,k}(t) = 2^{-j/2}\psi\left(2^{-j}t - k\right) \tag{3}$$

where $\psi_{j,k}(t)$ represents the discrete wavelet function at a specific scale j and position k. It is derived from the mother wavelet by scaling and translating it discretely, forming a set of orthogonal basis functions used in the discrete wavelet transform. t is the time variable, representing the domain over which the signal and wavelet function are defined. $\psi\left(2^{-j}t - k\right)$ is the scaled and shifted version of the mother wavelet function. Here, 2^{-j} scales the wavelet, compressing it for finer details (higher frequencies) or stretching it for broader features (lower frequencies), while $-k$ shifts the wavelet along the time axis.

On the other hand, the DWT coefficients are represented by Equation (4):

$$W_{j,k} = W\left(2^j, k2^j\right) = 2^{-j/2}\overline{\int_{-\infty}^{\infty} f(t)\psi(2^{-j}t - k)dt} \tag{4}$$

where $W_{j,k}$ represents the discrete wavelet transform coefficients at scale j and position k. $f(t)$ is the original function being analyzed. $\psi\left(2^{-j}t-k\right)$ is the scaled and shifted version of the mother wavelet function. $\int_{-\infty}^{\infty} \mathrm{f(t)}\psi(2^{-j}t-k)\Big)dt$ computes the projection of the signal $f(t)$ onto the wavelet function.

The simplest wavelet is the Haar wavelet, which is represented by a step function as shown in Equation (5):

$$\psi(t) = \begin{cases} 1, & 0 \le t < \frac{1}{2} \\ -1, & \frac{1}{2} \le t < 1 \\ 0, & else \end{cases} \tag{5}$$

The discrete Haar wavelet transform is widely used due to its simplicity and has been enhanced for various applications [10]. The process involves applying an ordered fast form of the transform to analyze a discrete signal. It starts with a one-dimensional array of 2^n entries and then undergoes n iterations of the same basic transform. This transform calculates a sample using the average and the difference between two points of an approximation function.

Before the iteration number l where $l \in \{1, ..., n\}$, this array consists of $2^{n-(l-1)}$ step-functions defined by (6) or (7):

$$\varphi_k^{(n-l)}(r) := \varphi_{[0,1]}\left(2^{n-l}\left[r - k2^{l-n}\right]\right) \tag{6}$$

$$\varphi_k^{(n-l)}(r) := \begin{cases} 1 & \text{if } k2^{l-n} \le r < (k+1)2^{l-n} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

where $\varphi_k^{(n-l)}(r)$ represents the scaling function, or approximation function, at the $(n-l)$ level of decomposition and position k. n is the total number of decomposition levels in the discrete wavelet transform. l is the current iteration or level of decomposition, where $l \in \{1, 2, ..., n\}$. As l increases, the decomposition goes deeper, capturing coarser details of the signal. $\varphi_{[0,1]}$ represents the basic scaling function defined over the interval $[0, 1]$, which is scaled and translated to different positions and resolutions.

After iteration l, the array will have half as many 2^{n-l} coefficients of 2^{n-l} step functions $\varphi_k^{(n-l)}$ and 2^{n-l} coefficients given by (8) or (9):

$$\psi_k^{(n-l)}(r) := \psi_{[0,1]}\left(2^{n-l}\left[r - k2^{l-n}\right]\right) \tag{8}$$

$$\psi_k^{(n-l)}(r) := \begin{cases} 1 & \text{if } k2^{l-n} \le r < (k+1)2^{l-n} \\ -1 & \text{if } \left(k + \left\lfloor \frac{1}{2} \right\rfloor\right)2^{l-n} \le r < (k+1)2^{l-n} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The calculation of the two wavelet coefficients, also called approximation coefficients and detail coefficients, in each iteration for an array of $2^{n-(l-1)}$ values is given by (10) and (11):

$$a_k^{(n-l)} := \frac{a_{2k}^{(n-[l-1])} + a_{2k+1}^{(n-[l-1])}}{2} \tag{10}$$

$$c_k^{(n-l)} := \frac{a_{2k}^{(n-[l-1])} - a_{2k+1}^{(n-[l-1])}}{2} \tag{11}$$

where $a_k^{(n-l)}$ represents the approximation coefficient at level $(n-l)$ and position k. These coefficients capture the low-frequency (smooth) components of the signal at a particular level of decomposition. $a_{2k}^{(n-[l-1])}$ and $a_{2k+1}^{(n-[l-1])}$ are the approximation coefficients from the previous level of decomposition, which are used to calculate the new approximation coefficient at the current level. $c_k^{(n-l)}$ represents the detail coefficient at level $(n-l)$ and

position k. These coefficients capture the high-frequency (detailed) components of the signal, such as sharp changes or edges, at a given level of decomposition.

The 2^{n-l} pairs of new coefficients constitute two arrays given by (12) and (13):

$$a^{(n-l)} := \left(a_0^{(n-l)}, a_1^{(n-l)}, \ldots, a_k^{(n-l)}, \ldots, a_{2^{(n-l)}-1}^{(n-l)}\right) \tag{12}$$

$$c^{(n-l)} := \left(c_0^{(n-l)}, c_1^{(n-l)}, \ldots, c_k^{(n-l)}, \ldots, c_{2^{(n-l)}-1}^{(n-l)}\right) \tag{13}$$

This algorithm allows the preservation of the basic information of the whole array.

1.1.3. Anomaly Behavior Analysis

Current cybersecurity solutions must be more effective to cope with the exponential increase in the quantity and complexity of cyber-attacks [8,22–25]. Two important techniques for detecting such threats are signature-based and anomaly-based Intrusion Detection Systems (IDS) [23,26,27]. A signature-based IDS relies on a set of known attack signatures or identities. However, these systems fail when it comes to detecting new attack types or even known attacks with small modifications to their base signatures. On the other hand, anomaly-based detection approaches excel in identifying novel and emerging threats or failures.

An anomaly-based IDS establishes a baseline model of the system's normal behavior through offline training (under known conditions) and flags any activity that deviates from this model as abnormal [28–30]. Configuration, misuse, or any fault can lead to abnormal behavior. However, this approach may generate numerous false alarms, which is a significant disadvantage.

1.1.4. Quality Control

Quality engineering ensures that projects, products, or services meet specified standards. It includes monitoring, testing, control, and taking corrective actions to identify and rectify defects or deviations from quality criteria. The objective is to produce reliable and accurate results, minimize issues, and continually enhance the quality of a given output [31]. Usually, quality control is linked with the information provided by a device or manually by humans. Using multiple sensors in industrial environments could benefit quality control in production lines. These benefits include the use of data analytics to detect possible issues in the whole process, the simulation of physical production through real-time data, and higher levels of worker engagement [31].

This work uses quality control techniques to inspect any sensor deviation. After obtaining the limits of normal operation, the samples from a sensor's wavelets are inspected to determine whether they are exhibiting normal behavior [32].

This paper is organized as follows: Section 1 provided a theoretical background on IoT applications, sensor faults, quality control, anomaly behavior analysis, and wavelets. Section 2 details the proposed methodology for monitoring sensor functionality. Section 3 presents the experimental results, and Section 4 offers conclusions and future work directions.

2. Materials and Methods

The methodology is divided into two phases: (A) offline and (B) online. The reference model is created offline, whereas the data obtained are compared against the data structure online. Figure 1 shows a general diagram of the proposed methodology.

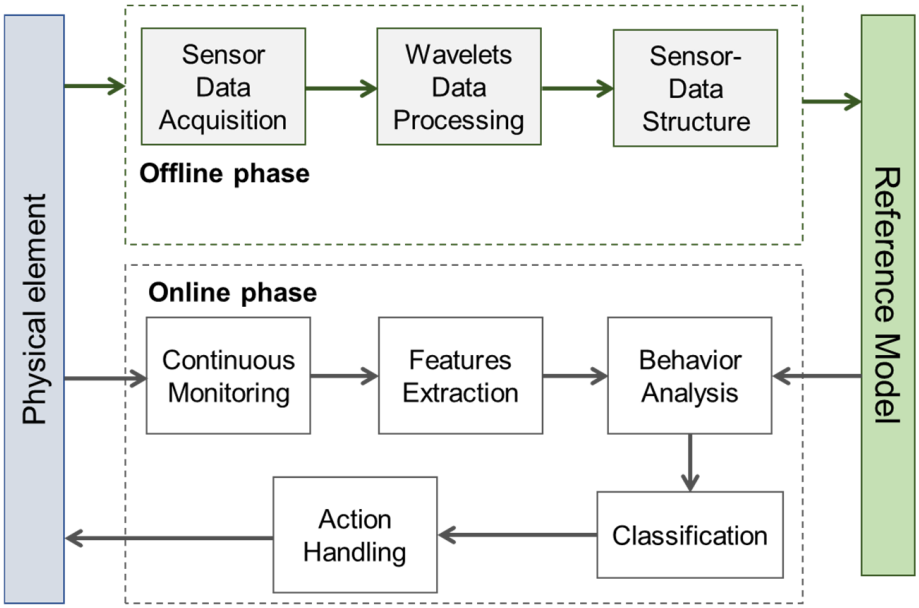

Figure 1. General diagram for the proposed methodology.

2.1. Offline Phase

During the data collection process, the sensors are connected to a custom-made embedded system board equipped with an STM32F411CEU microcontroller (see Figure 2). The STM32F411CEU, featuring a 100 MHz ARM Cortex-M4 core and a Floating-Point Unit (FPU), is well-suited for handling the computational demands of real-time wavelet transforms. The microcontroller is programmed to periodically sample the soil moisture sensor, capturing data at a rate of eight readings every 60 s. The analog signals from the sensors are converted into 10-bit digital values that represent the soil moisture percentage. Once the data are collected, they are processed using the DWT method. This involves decomposing the sensor data into different frequency components, as detailed in Equations (1)–(5). At each level of decomposition, the DWT extracts coefficients that are then used to construct a reference model, which serves as the basis for anomaly detection.

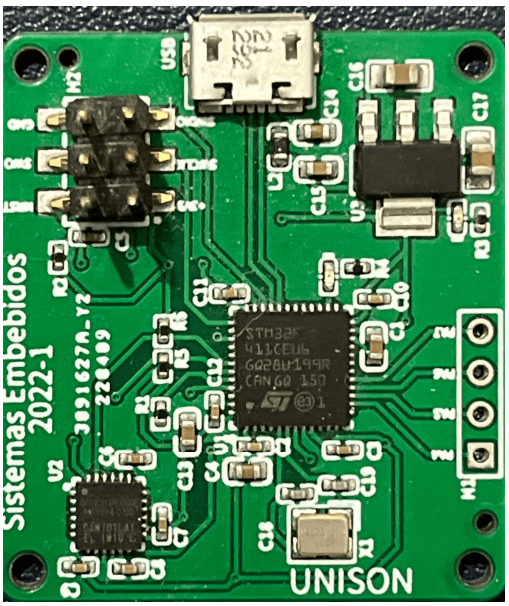

Figure 2. Custom-made embedded system.

The STM32F411's advanced processing capabilities and integrated peripherals facilitate efficient real-time analysis of the decomposed data. The FPU enhances the microcontroller's performance in performing the complex mathematical operations required for the DWT, ensuring that the system can accurately identify any deviations from normal sensor behavior.

One of the key features of the STM32F411CEU is its wide range of peripherals, including high-resolution ADCs that enable precise sensor data acquisition. It also offers flexible communication interfaces such as SPI, I2C, and UART, facilitating integration with sensors and other system components. These features make the STM32F411 ideal for applications requiring efficient signal processing and high peripheral integration.

In the offline phase, the approximation and detail coefficients for $n = 3$ are obtained for the signal array $x[2^n]$, which stores the 8 most recent moisture values ($2^n = 8$ incoming values) obtained from the sensor. Both coefficients are calculated through (10) and (11). After decomposing the signal, the coefficients from each level are combined into a 1-D array, which serves as the data structure.

The reference model is built offline using normal measurement attributes, specifically normal Euclidean distances, for each sensor data structure as described in Equation (14). To establish the control limits for normal operation, 10 arrays were utilized, each consisting of 8 data samples. These arrays provide a comprehensive representation of normal behavior by sequentially comparing each array with the remaining ones, resulting in a total of 45 Euclidean distance calculations (Euclidean samples, ES) to define the control limits. This approach ensures that every array contributes to a robust comparison framework, capturing the variability within normal operation. According to [32], using ten samples is sufficient for assessing deviations from nominal values within a normally distributed population, providing a solid basis for the control limit determination.

$$|e(r,s)| = \sqrt{\sum_{i=1}^{10}(s_i - r_i)^2} \tag{14}$$

In Equation (14), $|e(r,s)|$ represents the Euclidean distance between two vectors r and s. This distance measures the similarity or difference between the wavelet-transformed data of the current sample and the reference model, helping to identify deviations indicative of anomalies. s_i represents an element of any of the ten sample DWTs, while r_i represents an element of the reference DWT, obtained after applying the DWT to the first sample.

The reference model is then built with information about the control limits and the reference vectors. The control limits are computed as described in (15), where $\bar{e}$ is the mean value of the Euclidean distances and σ represents the standard deviation of the sample:

$$\bar{e} - 3\sigma \leq e(r,s) \leq \bar{e} + 3\sigma. \tag{15}$$

The reference model uses the 3 Sigma rule from [32] to establish control limits for detecting anomalies. By setting limits at three standard deviations above and below the mean of Euclidean distances from normal data, the model captures 99.73% of expected values, distinguishing normal variations from potential faults. A reference vector represents typical sensor behavior, and incoming data are compared against this baseline. Deviations beyond the control limits indicate anomalies, allowing the system to effectively identify and respond to faults, enhancing the accuracy and reliability of anomaly detection in embedded systems.

2.2. Online Phase

To illustrate the proposed approach in a real-world scenario, we present a case study focused on monitoring soil moisture levels in a Coriandrum sativum (coriander) plant. The objective of this experiment is to maintain the soil moisture within a predefined acceptable range with minimal variations. The experimental setup, as illustrated in Figure 3, involves

a soil moisture sensor that continuously measures the soil's moisture content. The sensor data are analyzed in real-time using the embedded Discrete Wavelet Transform (DWT) approach to detect any anomalies (which will be explained in Section 2.2.3). An actuator, represented by a valve, adjusts the watering based on the sensor's data to maintain optimal moisture levels. The system also includes two LEDs: a green LED that lights up to indicate normal behavior and a red LED that signals when abnormal conditions are detected. This setup enables continuous monitoring and adjustment of soil moisture while providing immediate visual feedback on the system's status.

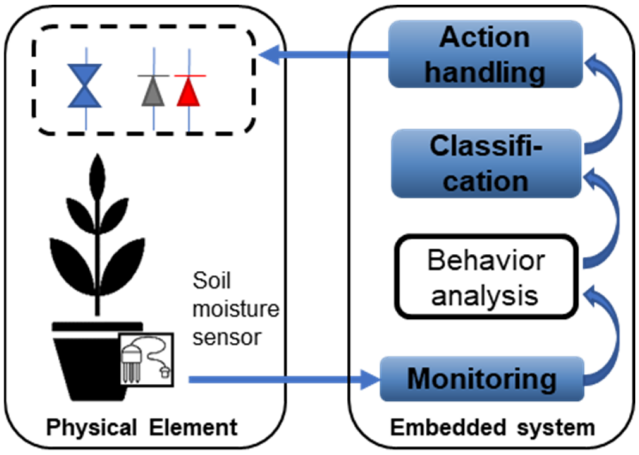

Figure 3. The architecture applied for the measurement of soil moisture in a coriander plant.

2.2.1. Continuous Monitoring

During the monitoring process, the microcontroller is programmed to collect soil moisture level data. It records eight soil moisture readings every 60 s (eight samples per minute). This approach is particularly feasible for resource-constrained embedded systems, allowing efficient implementation of the proposed methodology. The soil moisture sensor captures analog data, which the microcontroller converts into a 10-bit digital value representing the soil moisture percentage. This digital value is then processed and analyzed in real-time by the microcontroller to ensure accurate monitoring and timely adjustments of soil moisture levels.

2.2.2. Feature Extraction

In this stage, the data are processed similarly to wavelet data in the offline phase. The data are decomposed using a high-pass filter, aggregating the features into a 1-D vector. This new vector will aid in inspecting the sensor's behavior by comparing it with the reference model.

2.2.3. Behavior Analysis and Classification

The 1-D vector is compared with the representative vector in the reference model to obtain a Euclidean Distance. The same applies to the following 10 vectors to obtain the Euclidean distances. If any of the two conditions in (15) is met for (14) in any of the 10 distances, then the system is not behaving as expected, and the most probable cause is the sensor. If any conditions in (15) are not met, the data are classified as corrupt (wrong or harmful), then two actions will be triggered. The first action is conducted to stop the proliferation of the data, and the second is activating a visual alert.

2.2.4. Action Handling

The final stage involves activating one of two indicator LEDs based on the action determined by the classification unit. A green LED is turned on to indicate normal sensor functioning, while a red LED is activated to indicate a sensor fault. If a sensor fault is

detected, the microcontroller will close the valve and send a message indicating the need to repair or replace the sensor.

3. Results

The system architecture illustrated in Figure 4 was utilized to implement the proposed approach. It is a basic communication network consisting of three primary components: a computer, an embedded system, and a set of sensors and actuators.

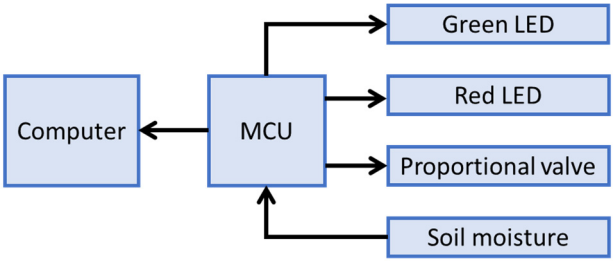

Figure 4. System architecture.

Figure 5 shows the testbed for the proposed approach. It includes a 32-bit microcontroller embedded in a custom-made board, which is connected to a soil moisture sensor using the sensor pinout: the analog output (AO), the ground (GND), and the voltage input (VCC). Pin A0 in the microcontroller was set as an analog input, while pins D1 and D0 were set as TX and RX connectivity pins, respectively. The sensor data were received using Algorithm 1, executed in a loop.

Algorithm 1: Transmitting data through microcontroller

Input: Raw analog values coming from sensor.
Output: Soil moisture percentage.
1. **for** k = 1 **to** 32 **do**
2. Analog-to-digital conversion of input
3. Digital value is stored as part of an array.
4. **end for**
5. **for** k = 1 to 32 do
6. Print value stored in the array.
7. **end for**

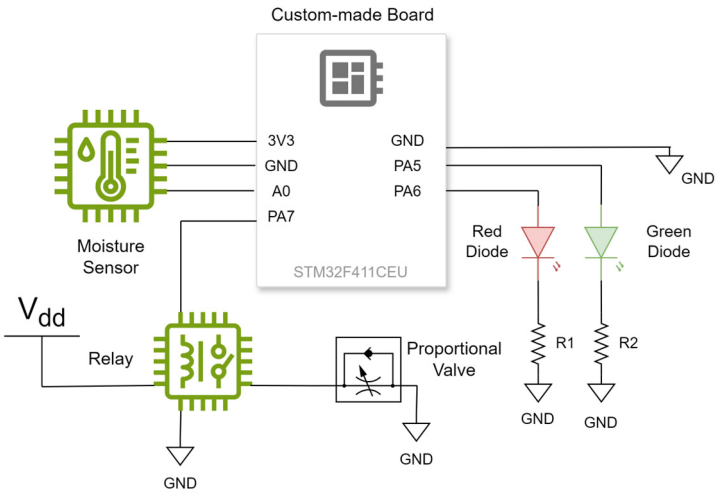

Figure 5. Schematic diagram of the testbed.

In Algorithm 1, incoming data are converted from analog to digital values and stored in a 32-element array. After that, these 32 elements are printed from the first to the last. The printed values are then received and plotted using Algorithm 2.

Algorithm 2: Receiving data from microcontroller

Input: Printed digital value from 0 to 1023.
Output: Plotted soil moisture values.
1. **for** k = 1 to 32 **do**
2. Convert digital value to soil moisture percentage.
3. **end for**
4. plot values

Algorithm 2 receives and splits the array the microcontroller sends into 32 values. Each value is multiplied by (100/1023) and subtracted from 100. This process converts each value from 0 to 1023 into a moisture percentage ranging from 0% to 100%. In this conversion, 100% represents the maximum moisture, and 0% represents the minimum moisture. Figure 6 displays the plot created using Algorithm 2. This plot illustrates normal moisture levels measured immediately after moderate plant watering.

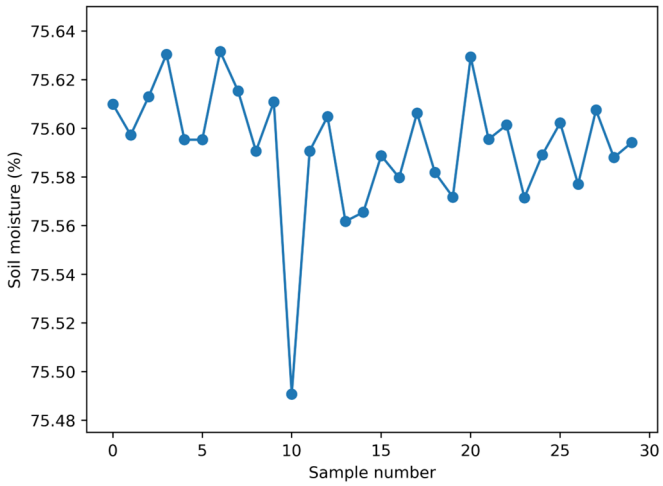

Figure 6. Soil moisture percentages obtained from an array.

Also, to evaluate the sensor behavior, the Euclidean distances of the Haar DWT were calculated and plotted for ten 1-D 8-element arrays. Algorithm 3 summarizes the steps used to obtain these distances. This process involves creating a 1-D wavelet pattern and a set of additional wavelets of the same dimension.

Figure 7 depicts a plot of the 30 Euclidean distances calculated using Algorithm 3. In the plot, all the values fall within the range defined by $\overline{x} \mp 3\sigma$ (green and blue lines), referred to as control limits: Upper Control Limit (UCL) and Lower Control Limit (LCL). This outcome suggests that the sensor accurately measures stable moisture levels, indicated by the activation of a green LED. However, if the results fall outside these control limits, a red LED is activated, and the plant watering valve will close to allow for sensor replacement or repair. These procedures are summarized in Algorithm 4.

Algorithm 3: Computing of Euclidean distances between DWTs

Input: Soil moisture percentage
Output: Euclidean distance between pattern and calculated wavelets
1. **for** k = 1 to 8 **do**
2. A soil moisture value is stored in an 8-element array.
3. **end for**
4. Approximation and detail coefficients are calculated for the pattern wavelet.
5. **for** k = 1 to 10 **do**
6. **for** k = 1 to 8 **do**
7. A soil moisture value is stored in an 8-element array.
8. **end for**
9. Approximation and detail coefficients are calculated for the wavelet.
10. Euclidean distance between the pattern and current wavelet is calculated.
11. Euclidean distance is printed.
12. **end for**

Algorithm 4: Sensor element activation

Input: Euclidean distance between wavelets.
Output: Signal to activate led and modify valve closing.
1. **if** $\overline{x} - 3\sigma <$ Euclidean distance $< \overline{x} + 3\sigma$
2. turn green LED on
3. **else if** Euclidean distance $> \overline{x} + 3\sigma$ or Euclidean distance $> \overline{x} - 3\sigma$
4. turn red LED on
5. close valve
6. **end if**

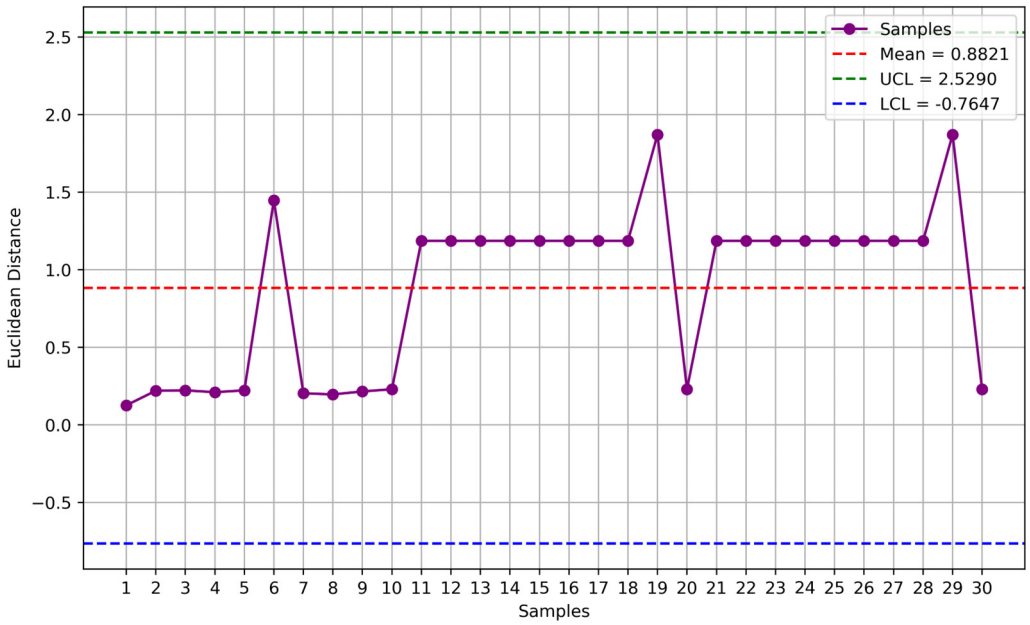

Figure 7. Euclidean distances between the baseline wavelet and the online wavelets.

Figure 7 illustrates that all recent measurements fall within the three-sigma limits from the mean, indicating minimal deviation from the mean and, thus, the correct functioning of the sensor.

Once normal behavior has been identified, the next step is to manipulate the sensor to confirm its capability to detect issues.

Figure 8 provides a crucial evaluation of the wavelet-based anomaly detection system by comparing normal and abnormal sensor behaviors using Euclidean distance metrics derived from DWT coefficients. The figure illustrates 16 data points out of control limits.

It also illustrates the differences in Euclidean distances between the wavelet-transformed data of normal sensor operations and those subjected to induced perturbations, simulating real-world faults. As can be seen, the Euclidean distances for normal sensor behavior are shown to consistently fall within the established control limits, typically defined as the mean ($\overline{x}$) plus or minus three standard deviations ($\mp 3\sigma$). These control limits act as thresholds to distinguish between normal and abnormal operations. The data points representing normal behavior remain well within these boundaries, indicating stable sensor performance under standard conditions. Conversely, the abnormal data points, which are introduced through perturbations such as altering the sensor's position or adding external noise, exhibit significant deviations from the normal range. These deviations result in Euclidean distances that fall outside the control limits, effectively triggering the anomaly detection mechanism. This separation between normal and abnormal data points underscores the system's sensitivity to deviations and its capability to detect anomalies, even when the perturbations are relatively subtle. The distinct separation of normal and abnormal behaviors in the Euclidean distance space highlights the system's high sensitivity and specificity in anomaly detection. The abnormal points show a wide spread beyond the control limits, suggesting that the system is capable of distinguishing various types of anomalies, such as gradual drifts and abrupt faults. This behavior pattern analysis is valuable, as it allows the system to classify the nature and severity of sensor faults, enhancing its fault isolation capabilities.

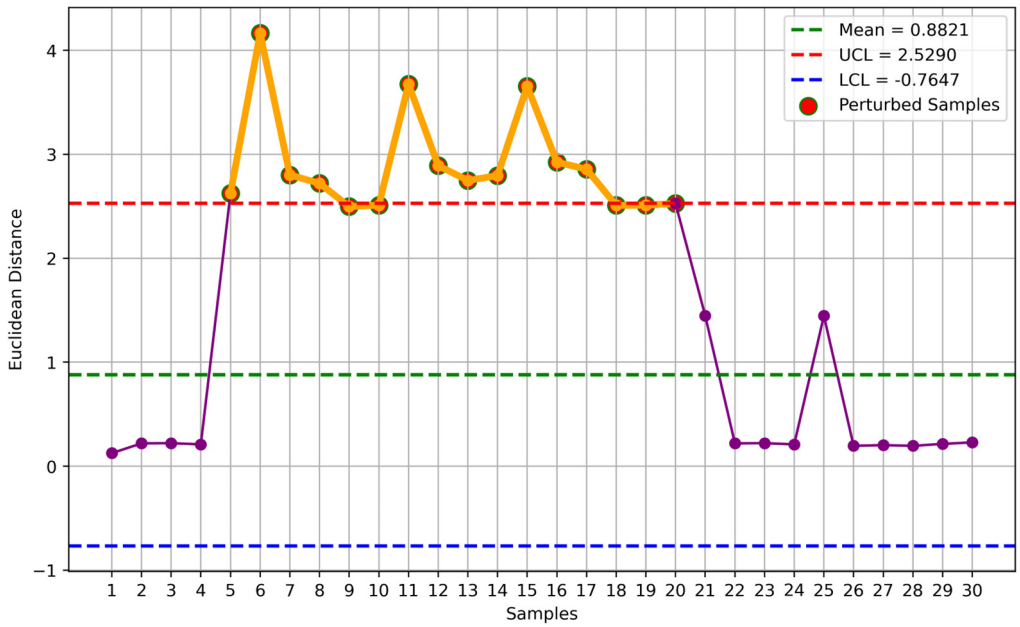

Figure 8. Comparison between normal and abnormal behavior.

The implications of these findings for real-time monitoring are significant. The wavelet-based approach demonstrated in Figure 8 allows for the rapid detection of deviations, enabling timely responses to potential faults in embedded systems. This capability is essential for maintaining system integrity and preventing erroneous data propagation, which could lead to broader operational failures. The results confirm that the wavelet-based method is robust and effective for real-time anomaly detection, making it a valuable tool for enhancing the reliability and security of embedded sensor systems.

To compare the presented method against others in the literature, false sensor data injection was implemented by altering the moisture sensor's signal to simulate erroneous data, thereby challenging the system's ability to distinguish between genuine and manipulated sensor outputs [33]. This procedure involved injecting false data directly into the sensor's analog signal path using a variable resistor or an external signal generator to su-

perimpose incorrect readings onto the actual sensor data. This method replicated common sensor faults, such as drift, bias, or abrupt deviations, providing a realistic scenario to assess the effectiveness of the anomaly detection framework. Across over 1000 datapoints of injection, the DWT combined with Euclidean distance metrics was particularly effective in detecting this false data injection, achieving a detection rate of up to 93%, capturing both transient and persistent characteristics of the injected false data. Considering the metrics introduced in [33], where the best-case scenario for false sensor data injection reached a 96% detection rate, the results appear to outperform our proposed approach. However, the high accuracy was achieved using a Support Vector Machine, which is not optimized for resource-constrained environments such as the STM32F411. Additionally, DWT's computational efficiency makes it ideal for real-time monitoring on resource-constrained embedded systems, ensuring the prompt detection and isolation of faults or false data injections.

After conducting a thorough review of the state of the art in the field of anomaly detection, specifically in applications that are implementable and applicable in embedded systems, we found a limited number of works that directly address this area. Most of the existing research either focuses on computationally expensive algorithms or simulation-based studies that are not directly comparable to our approach, which emphasize real-time implementation in resource-constrained environments.

However, four references were identified [34–37], that are closely related to our proposed approach. To aid in making a meaningful comparison, a comparison matrix (see Table 1) was designed to contrast our approach with these selected studies using both qualitative and quantitative characteristics. The comparison addresses aspects such as main focus, anomaly detection methods, detection accuracy, resource consumption, control strategy, and real-time applicability. This structured comparison provides insights into how our approach differs and highlights the novelty and practicality of our system for real-time anomaly detection in embedded systems. In Table 1, the following acronyms are used:

- DWT: Discrete Wavelet Transform.
- CNN: Convolutional Neural Network.
- PCA: Principal Component Analysis.
- WNN: Wavelet Neural.
- ML: Machine learning.
- ED: Euclidean distance.

Based on Table 1, much of the current literature on anomaly detection relies on simulations or computationally intensive algorithms that are hard to validate in real-world applications due to their complex nature. These approaches often require significant industrial infrastructure changes, which are not practical or recommended for existing processes. On the other hand, our approach focuses on lightweight computational solutions that can be implemented on readily available yet powerful embedded systems, such as 32-bit microcontrollers.

Using 32-bit microcontrollers and a custom-based PCB design, the proposed approach reduces computational overhead and facilitates seamless integration into existing industrial processes. This makes deploying and evaluating our system in real-world settings significantly easier, avoiding the limitations of solutions confined to research platforms or simulations.

In the anomaly detection community context, the proposed wavelet anomaly detection scheme represents an alternative to deploying anomaly detection solutions. It emphasizes real-world applicability by providing a solution that can be deployed in real-time on actual industrial systems rather than remaining theoretical or experimental. This practical focus distinguishes our work and addresses a gap in the literature where many solutions are not easily translatable to operational environments.

Table 1. Comparison matrix between the proposed scheme and existing related works.

Dimension	Proposed Approach	Reference [37]	Reference [36]	Reference [35]	Reference [34]
Main Focus	Real-time anomaly detection using DWT.	Detection defects using Eddy Current and CNN.	Crack detection using PCA and wavelet denoising.	Anomaly detection using WNN.	Detection and classification in power systems using DWT and ML.
Anomaly Detection Method	DWT with ED for anomaly detection.	Sensors with CWT and CNN for feature extraction.	Adaptive PCA combined with wavelet denoising.	WNN with function-aware feature extraction.	DWT for feature extraction and ML classifiers.
Detection Accuracy	Up to 93% detection.	Up to 98% classification.	False Alarm Rate < 3%.	91.17% average detection.	100% classification.
Real-time Applicability	Anomaly detection in embedded systems.	Detection and classification in rail inspection systems.	Monitoring of industrial systems for crack detection.	Anomaly detection in industrial communication.	Voltage sag detection in power systems.
Resource Consumption	Low, optimized for embedded systems.	Higher due to CNN processing and multiple sensor inputs.	Requires higher computational power for PCA and denoising.	Higher due to WNN and function-aware anomaly detection.	Higher due to machine learning classifiers and large datasets.
Control Strategy	Simple upper and lower limits based on three standard deviations (3σ).	Classification of defects using CNN.	Adaptive control with PCA-based thresholding.	Function-aware anomaly detection using WNN.	Classification of normal vs. anomaly events using ML classifiers.

4. Discussion and Conclusions

In this paper, we have demonstrated that it is highly effective to continuously monitor the behavior of a soil moisture sensor by employing advanced techniques such as discrete wavelet transform (DWT) together with Euclidean distances. These methods enable us to detect subtle changes in sensor performance, ensuring that any deviations from optimal functionality are identified and addressed. By maintaining the sensor's accuracy and reliability, we can provide a precise and responsive system that consistently meets the moisture needs of plants.

Compared with machine learning techniques like SVMs and neural networks, which require extensive computational resources, DWT+ED is computationally efficient and suitable for real-time applications on resource-constrained devices like microcontrollers. The Haar wavelet allows for fast signal decomposition, and the simple calculation of Euclidean distance quantifies deviations effectively, achieving up to a 93% detection accuracy. This balance of performance and low computational overhead highlights the novelty and practicality of the DWT+ED approach for embedded systems.

It is also important to explore the integration of the wavelet-based anomaly detection system into diverse real-world scenarios beyond the current setup. Potential applications include industrial automation, where real-time fault detection in critical sensors could prevent costly downtime, and smart healthcare, where monitoring patient vitals with embedded sensors could enhance patient safety by promptly identifying anomalies. Additionally, expanding the system's capabilities to handle multi-sensor environments would significantly enhance its robustness, allowing it to process data from various sources such as temperature, humidity, or pressure sensors simultaneously. For future research, exploring the use of advanced wavelet transforms, such as continuous wavelet transforms (CWT) or adaptive wavelets tailored to specific signal characteristics, could further improve the detection accuracy and adaptability of the system.

While exploring the integration of machine learning techniques, such as neural networks or reinforcement learning, offers promising avenues for dynamically adjusting detection thresholds and enhancing decision-making under evolving conditions, further

experimental validation is crucial for a comprehensive evaluation. There is an open area to conduct additional experiments such as injecting faults, tampering with sensors, or executing sophisticated cyber-attacks like Denial of Service (DoS) and impersonation attacks. Future work should aim to include these types of tests to fully assess the robustness and applicability of the proposed approach in real-world scenarios, thereby further strengthening its performance and resilience in complex, dynamic environments.

Author Contributions: Conceptualization, J.P. and V.H.B.; Data curation, G.P.; Formal analysis, J.P. and G.P.; Funding acquisition, A.B.; Investigation, J.P. and G.P.; Methodology, J.P.; Project administration, J.P.; Resources, J.P., V.H.B. and A.B.; Software, G.P.; Supervision, J.P.; Validation, J.P. and G.P.; Visualization, G.P.; Writing—original draft, J.P. and V.H.B.; Writing—review and editing, J.P., V.H.B. and A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: This work was supported by the department of Industrial Engineering at Universidad de Sonora.

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

References

1. Li, D.; Wang, Y.; Wang, J.; Wang, C.; Duan, Y. Recent Advances in Sensor Fault Diagnosis: A Review. *Sens. Actuators A Phys.* **2020**, *309*, 111990. [CrossRef]
2. Fatima, N.; Riaz, S.; Ali, S.; Khan, R.; Ullah, M.; Kwak, D. Sensors Faults Classification and Faulty Signals Reconstruction Using Deep Learning. *IEEE Access* **2024**, *12*, 100544–100558. [CrossRef]
3. Yang, J.W.; Lee, Y.D.; Koo, I.S. Convolutional Autoencoder-Based Sensor Fault Classification. *Int. Conf. Ubiquitous Future Netw.* **2018**, *2018*, 865–867. [CrossRef]
4. Jiang, X.; Zhang, X.; Zhang, Y. Establishment and Optimization of Sensor Fault Identification Model Based on Classification and Regression Tree and Particle Swarm Optimization. *Mater. Res. Express* **2021**, *8*, 085703. [CrossRef]
5. Jiang, C.Y.; Li, L.C.; Ye, C.L.; Yu, S.Y. Research on Sensor Fault Identification Based on Improved 1-v-r SVM Classification Method. *Int. J. Adv. Media Commun.* **2016**, *6*, 235–245. [CrossRef]
6. Sun, Y.; Liu, S.; Yu, Y.; Zhao, T.; Zou, Z.; Zhang, J.; Zhang, S.; Zhang, H. Gas Sensor Fault Diagnosis for Imbalanced Data Based on Generative Adversarial Networks. *J. Phys. Conf. Ser.* **2021**, *1732*, 012033. [CrossRef]
7. Huang, J.; Li, M.; Zhang, Y.; Mu, L.; Ao, Z.; Gong, H. Fault Detection and Classification for Sensor Faults of UAV by Deep Learning and Time-Frequency Analysis. In Proceedings of the 2021 40th Chinese Control Conference (CCC), Shanghai, China, 26–28 July 2021; Volume 2021, pp. 4420–4424. [CrossRef]
8. Pacheco, J.; Benitez, V.H.; Felix-Herran, L.C.; Satam, P. Artificial Neural Networks-Based Intrusion Detection System for Internet of Things Fog Nodes. *IEEE Access* **2020**, *8*, 73907–73918. [CrossRef]
9. Rhif, M.; Ben Abbes, A.; Farah, I.R.; Martínez, B.; Sang, Y. Wavelet Transform Application for/in Non-Stationary Time-Series Analysis: A Review. *Appl. Sci.* **2019**, *9*, 1345. [CrossRef]
10. Guo, T.; Zhang, T.; Lim, E.; Lopez-Benitez, M.; Ma, F.; Yu, L. A Review of Wavelet Analysis and Its Applications: Challenges and Opportunities. *IEEE Access* **2022**, *10*, 58869–58903. [CrossRef]
11. Verma, S.; Sahu, S.P.; Sahu, T.P. Discrete Wavelet Transform-Based Feature Engineering for Stock Market Prediction. *Int. J. Inf. Technol.* **2023**, *15*, 1179–1188. [CrossRef]
12. Perez, G.; Pacheco, J.; Benitez, V. Anomaly Behavior Analysis for Sensors Fault Detection. In Proceedings of the 2023 IEEE Symposium Series on Computational Intelligence, SSCI 2023, Mexico City, Mexico, 5–8 December 2023; pp. 1718–1723. [CrossRef]
13. Babu Rao, K.; Mallikarjuna Reddy, D. Fault Detection in Rotor System by Discrete Wavelet Neural Network Algorithm. *J. Vib. Control.* **2021**, *28*, 3315–3331. [CrossRef]
14. Almounajjed, A.; Sahoo, A.K. Wavelet-Based Multi-Class Support Vector Machine for Stator Fault Diagnosis in Induction Motor. *Trans. Inst. Meas. Control* **2022**, *45*, 261–273. [CrossRef]
15. Lee, X.Y.; Kumar, A.; Vidyaratne, L.; Rao, A.R.; Farahat, A.; Gupta, C. An Ensemble of Convolution-Based Methods for Fault Detection Using Vibration Signals. In Proceedings of the 2023 IEEE International Conference on Prognostics and Health Management, ICPHM 2023, Montreal, QC, Canada, 5–7 June 2023; pp. 1718–1723. [CrossRef]
16. Mandal, S.; Santhi, B.; Sridhar, S.; Vinolia, K.; Swaminathan, P. A Novel Approach for Fault Detection and Classification of the Thermocouple Sensor in Nuclear Power Plant Using Singular Value Decomposition and Symbolic Dynamic Filter. *Ann. Nucl. Energy* **2017**, *103*, 440–453. [CrossRef]

17. Troni, G.; Whitcomb, L.L. Field Sensor Bias Calibration with Angular-Rate Sensors: Theory and Experimental Evaluation with Application to Magnetometer Calibration. *IEEE/ASME Trans. Mechatron.* **2020**, *24*, 1698–1710. [CrossRef]
18. Jihani, N.; Kabbaj, M.N.; Benbrahim, M. Sensor Fault Detection and Isolation for Smart Irrigation Wireless Sensor Network Based on Parity Space. *Int. J. Electr. Comput. Eng. (IJECE)* **2023**, *13*, 1463–1471. [CrossRef]
19. Han, X.; Jiang, J.; Xu, A.; Bari, A.; Pei, C.; Sun, Y. Sensor Drift Detection Based on Discrete Wavelet Transform and Grey Models. *IEEE Access* **2020**, *8*, 204389–204399. [CrossRef]
20. Pereira, M.; Glisic, B. Detection and Quantification of Temperature Sensor Drift Using Probabilistic Neural Networks. *Expert Syst. Appl.* **2023**, *213*, 118884. [CrossRef]
21. Daubechies, I. Orthonormal Bases of Compactly Supported Wavelets. *Commun. Pure Appl. Math.* **1988**, *41*, 909–996. [CrossRef]
22. Maseer, Z.K.; Yusof, R.; Bahaman, N.; Mostafa, S.A.; Foozy, C.F.M. Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset. *IEEE Access* **2021**, *9*, 22351–22370. [CrossRef]
23. Aldweesh, A.; Derhab, A.; Emam, A.Z. Deep Learning Approaches for Anomaly-Based Intrusion Detection Systems: A Survey, Taxonomy, and Open Issues. *Knowl.-Based Syst.* **2020**, *189*, 105124. [CrossRef]
24. Maseer, Z.K.; Yusof, R.; Al-Bander, B.; Saif, A.; Kadhim, Q.K. Meta-Analysis and Systematic Review for Anomaly Network Intrusion Detection Systems: Detection Methods, Dataset, Validation Methodology, and Challenges. *IET Netw.* **2023**. [CrossRef]
25. Pacheco, J.; Hariri, S. Anomaly Behavior Analysis for IoT Sensors. *Trans. Emerg. Telecommun. Technol.* **2018**, *29*, e3188. [CrossRef]
26. Idrissi, M.J.; Alami, H.; El Mahdaouy, A.; El Mekki, A.; Oualil, S.; Yartaoui, Z.; Berrada, I. Fed-ANIDS: Federated Learning for Anomaly-Based Network Intrusion Detection Systems. *Expert Syst. Appl.* **2023**, *234*, 121000. [CrossRef]
27. Kwon, H.Y.; Kim, T.; Lee, M.K. Advanced Intrusion Detection Combining Signature-Based and Behavior-Based Detection Methods. *Electronics* **2022**, *11*, 867. [CrossRef]
28. Bhavsar, M.; Roy, K.; Kelly, J.; Olusola, O. Anomaly-Based Intrusion Detection System for IoT Application. *Discov. Internet Things* **2023**, *3*, 5. [CrossRef]
29. Thakkar, A.; Lohiya, R. A Review on Challenges and Future Research Directions for Machine Learning-Based Intrusion Detection System. *Arch. Comput. Methods Eng.* **2023**, *30*, 4245–4269. [CrossRef]
30. Abdulganiyu, O.H.; Ait Tchakoucht, T.; Saheed, Y.K. A Systematic Literature Review for Network Intrusion Detection System (IDS). *Int. J. Inf. Secur.* **2023**, *22*, 1125–1162. [CrossRef]
31. Godina, R.; Matias, J.C.O. Quality Control in the Context of Industry 4.0. In Proceedings of the Industrial Engineering and Operations Management II: XXIV IJCIEOM, Lisbon, Portugal, 18–20 July 2019; Springer Proceedings in Mathematics & Statistics. Volume 281, pp. 177–187. [CrossRef]
32. Montgomery, D.C. *Introduction to Statistical Quality Control*, 8th ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2019; ISBN 978-1-119-39930-8.
33. Jane Nithya, K.; Shyamala, K. A Systematic Review on Various Attack Detection Methods for Wireless Sensor Networks. In *Proceedings of the International Conference on Innovative Computing and Communications: Proceedings of ICICC 2021*; Springer: Singapore, 2022; pp. 183–204.
34. Alshareef, S.M. Voltage Sag Assessment, Detection, and Classification in Distribution Systems Embedded with Fast Charging Stations. *IEEE Access* **2023**, *11*, 89864–89880. [CrossRef]
35. Wan, M.; Song, Y.; Jing, Y.; Wang, J. Function-Aware Anomaly Detection Based on Wavelet Neural Network for Industrial Control Communication. *Secur. Commun. Netw.* **2018**, *2018*, 5103270. [CrossRef]
36. Faizan-E-Mustafa; Ahmed, I.; Maaruf, M.; Khalid, M. Detection of Cracks in the Industrial System Using Adaptive Principal Component Analysis and Wavelet Denoising. In Proceedings of the IEEE International Conference on Industrial Technology, Bristol, UK, 25–27 March 2024. [CrossRef]
37. Alvarenga, T.A.; Carvalho, A.L.; Honorio, L.M.; Cerqueira, A.S.; Filho, L.M.A.; Nobrega, R.A. Detection and Clas-sification System for Rail Surface Defects Based on Eddy Current. *Sensors* **2021**, *21*, 7937. [CrossRef]