2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Anomaly Behavior Analysis for Sensors Fault Detection

Guillermo Pérez
Industrial Engineering
Universidad de Sonora
Hermosillo, Sonora
a222230081@ unison.mx

Jesus Pacheco
Industrial Engineering
Universidad de Sonora
Hermosillo, Sonora
jesus.pacheco@ unison.mx

Víctor Benitez
Industrial Engineering
Universidad de Sonora
Hermosillo, Sonora
victor.benitez@ unison.mx

Abstract— **In today's world, sensors play a crucial role, as they feed information to make accurate decisions and take actions; therefore, making sure that sensors behave correctly is critical. In this work, we focus on inspecting the data provided by sensors, aiming at discovering any issue due to malfunction, misuse, or any other source of error before the issue is propagated through the system. To achieve that, we propose a novel approach based on wavelets embedded in a microcontroller to analyze data from sensors. The objective is to determine whether the sensor is issuing correct data (normal behavior) or not (abnormal behavior), to prevent the error from reaching other parts of the system.**

Keywords— discrete wavelet transform, Euclidean distance, Embedded Systems, anomaly behavior analysis, sensor's fault detection.

I. Introduction

Effective communication is vital in delivering necessary information, particularly in critical situations requiring accurate actions. However, the inclusion of components like sensors introduces significant security challenges, as they expand the potential for errors through the systems, deterring the accurate delivery of information to end users. This paper introduces a methodology for constructing an Intrusion Detection System utilizing Anomaly Behavior Analysis to identify when an IoT network node with sensing capabilities is compromised. Our initial experimental results demonstrate the effectiveness of our approach in detecting both known and unknown anomalies caused by misuse, malfunction, or any other sources of sensor errors before the error is spread deep into the network. The experiments are conducted with the aid of an embedded system which represents an IoT node sending information through the network. The results indicate a high detection rate with low false alarms.

The paper is organized as follows. Section II gives a brief theoretical background about IoT applications, sensor fault, quality control, anomaly behavior analysis, and wavelets. Section III presents the methodology proposed to properly monitor the functioning of the sensor attached to an embedded system. Next, Section IV presents the experimental results of the proposed approach. Finally, section V provides conclusions and possible future works.

II. Background

With the exponential use of IoT applications (e.g., monitoring, guidance, premises control, etc.), ensuring the seamless operation of all interconnected devices becomes increasingly challenging. Primarily, the proper functioning of sensors stands as a crucial element in any IoT application, demanding ongoing research to detect and address device faults. Furthermore, safeguarding data integrity and mitigating potential risks emerge as significant concerns for companies, regular users, and researchers in this domain.

The following is a brief outline of the most common types of sensor fault, some of the strategies already being applied to tackle the issues, and the state of the art in data security preservation.

A. Sensor Fault Classification

According to Han et al., [1] sensor fault can be classified into two groups: 1) incipient fault, and 2) abrupt fault.

1) Incipient fault: An incipient fault is a small amount of abnormal or unstable working status that develops over time, causing more severe faults in the long term [1]. Common faults in this group include sensor bias and sensor drift.

When sensor bias occurs, data values are replaced with a constant value, causing a loss of accuracy in the device. Because it is a common fault, many solving strategies have been proposed. For example, in the branch of field sensors, Troni and Whitcomb [2] suggest using angular-rate-aided estimation methods that improve bias estimation in comparison with previous proposals by other authors. Similarly, Yang et al. [3] studied bias in accelerometers applied in the drilling process and proposed a sensor fault detection and isolation method that outperforms previous strategies.

Sensor drift means the presence of an offset or gain parameter that changes slowly over time. Han et al. [4] created a sensor drift detection method based on discrete wavelet transform (DWT) and grey models, with the former used to decompose the signal and the latter for detrending. Conversely, Pereira and Glisic [5] propose detecting and quantifying sensor drift in temperature sensors based on trinomial distribution, using probabilistic neural networks (PNN) to estimate temperature.

2) Abrupt fault: On the other hand, abrupt fault occurs when the sensor stops working suddenly because of physical damage, generating a fault parameter that is easy to identify [1]. This group includes sensor noise, short and open circuits, and sensor random faults. Sensor noise can appear in two forms: internal (from the sensor itself and its circuit) and environmental (from external interference). Short- and open-circuit faults are caused, respectively, by poor contact and disconnections. Finally, random sensor faults are caused by the complex environment of the layout, which can exceed the sensor requirements [1].

B. Wavelets

Wavelets are waveforms of limited duration used for analyzing functions. Moreover, they allow signals with localized features to be broken down into different frequency components at different scales. Wavelet analysis involves decomposing signals into wavelet coefficients, which represent their components at different scales and positions in time, in contrast to similar approaches such as the Fourier transform [6,7]. Because of these features, wavelets are widely applied when solving problems associated with time-varying non-stationary variables. Particularly, the wavelet transform allows the representation of a signal as a set of basic functions (wavelets) obtained from the translation and scalation of a mother wavelet given by (1):

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

The continuous wavelet transform (CWT) allows the mapping of the properties of non-stationary signals. In the time-frequency representation, the coefficients $W_f(a,b)$ in (2) are obtained through changes in the scale and the position parameters of a signal [7]:

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

There is also a discrete wavelet transform (DWT), which consists of the decomposition of the signal into a mutually orthogonal set of wavelets. This transform is expressed by (3):

$$\Psi_{j,k}(t) = 2^{-\frac{j}{2}}\psi\,(2^{-j}t - k) \tag{3}$$

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

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

The simplest wavelet is the Haar wavelet, a step function given in (5):

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

Because of its simplicity, it has been extensively applied and improved in different scenarios [6].

It is possible to apply an ordered fast form of the discrete Haar wavelet transform to analyze a discrete signal. This process begins with a one-dimensional array of 2^n entries, followed by n iterations of the same basic transform, which consists of the calculation of a sample given by two values, the average and the change between two points of an approximation function.

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

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

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

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 of wavelets given by (8) or (9):

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

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

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):

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

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

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

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

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

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

C. Anomaly Behavior Analysis

The current state of cyber-security solutions falls short of effectively countering the exponential increase in both the quantity and complexity of cyber-attacks [8], [9]. Two fundamental techniques for detecting such attacks are signature-based and anomaly-based Intrusion Detection Systems (IDS) [10], [11], [12]. Signature-based IDS relies on a database of known attack signatures or identities. However, these systems fail to detect new attack types or even known attacks with minor modifications to their signatures. On the other hand, anomaly detection approaches excel in identifying novel and emerging attacks.

Anomaly-based IDS establishes a baseline model of the system's normal behavior through offline training and flags any activity that deviates from this model as an anomaly [11],

[12]. Any attack, misconfiguration, or misuse will result in behavior that deviates from the norm, which we classify as abnormal behavior. However, a significant drawback of this approach is the potential for generating a large number of false alarms.

D. *Quality Control*

Quality control in engineering is used to ensure that projects, products, or services meet specified quality standards. It involves activities such as monitoring, testing, control, and corrective actions to identify and rectify defects or deviations from quality criteria. The goal is to produce reliable and accurate results, minimize issues, and continuously improve the quality of a given output [13]. 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 using data analytics to detect possible issues in the whole process, the simulation of physical production through real-time data, and more worker engagement [13].

In this work, quality control is used to inspect any deviation of the sensor. After obtaining the limits of normal operation, the samples from a sensor's wavelets are inspected to determine whether they are exhibiting normal behavior [14].

III. Methodology

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

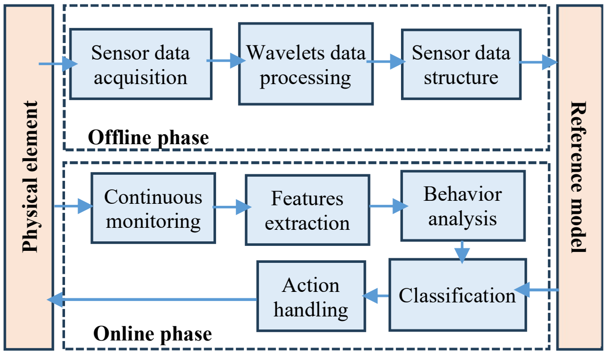

Fig. 1. General diagram for the proposed methodology.

A. *Offline Phase*

Sensors are directly connected to a NUCLEO-F411RE embedded system (Fig. 2). Data obtained from the sensor is decomposed using the DWT method as shown in Equations (1) to (5). In each level of the decomposition, the extracted coefficients are used to build a data structure (reference model).

For the first part, the approximation and detail coefficients for $n = 3$ are obtained for the signal array *x[8]*, which stores the 8 most recent moisture values obtained from the sensor. Both coefficients are calculated through (10) and (11). Once the signal is decomposed, the coefficients of each level are aggregated in a 1-D array, which is the data structure.

The reference model is then built offline by using normal measurement attributes (normal Euclidean distance) for each sensor data structure (see Equation (12)). Ten arrays are used to find the control limits for normal operation. Each array is compared with the rest having 80 Euclidean distances (referred as Euclidean samples, ES) to build control limits [13]. As mentioned in [14], 10 samples are enough to inspect deviation from nominal values in a normal distributed population.

$$e(r,s) = \sqrt{\sum_{i=1}^{8}(s_i - r_i)^2} \tag{12}$$

In (12), s_i represents an element of any of the 10 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 the information about the control limits and the reference vectors. The control limits are computed as described in (13):

$$\bar{e} - 3\sigma \leq e(r,s) \leq \bar{e} + 3\sigma \tag{13}$$

The reference model contains the control limits and a representative vector.

B. *Online Phase*

To illustrate the proposed approach in a real-world scenario, the soil moisture measurement in a Coriandrum sativum (i.e., coriander) plant is used. Aiming at guaranteeing soil moisture levels within an acceptable range and with smooth, minimal variations, the architecture depicted in Fig. 2 was applied, where the soil moisture sensor represents the sensor element, and the actuator element is represented by a valve that waters the plant and two diodes (green and red) are used to notify of normal or abnormal behavior, respectively.

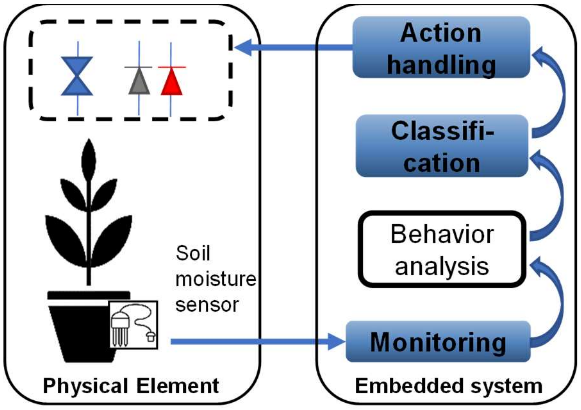

Fig. 2. Methodology applied for the measurement of soil moisture in a coriander plant.

1) Continuous monitoring: In the monitoring part, with the aid of a NUCLEO-F411RE embedded system, the sensor registers eight soil moisture level values every 60 seconds at 1 second sampling rate. The small number of samples (8 samples) will help to implement the proposed approach in resources-constrained embedded systems. The sensor accomplishes this recording by obtaining a digital value between 0 and 1023, converted into a soil moisture percentage.

2) Feature extraction: This stage is similar to the wavelet data processing function in the offline phase. The data obtained is decomposed using the high-pass filter level and aggregating the features in a 1-D vector. This new vector will help to inspect the behavior of the sensor by comparing it against the reference model.

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

4) the next 10 vectors, to obtain 10 Euclidean distances. If any of the two conditions in (13) is met for (12) in any of the 10 distances, then the system is not behaving as expected and the most probable cause is the sensor. In the case that any of the conditions in (13) is not met, the data is classified as corrupt (wrong or bad data), then two actions will be triggered. The first action is conducted to stop the proliferation of the data and the second one is the activation of a visual alert.

5) Action handling: With the plan of action already decided by the classification unit, the last stage consists of turning on one of two indicator LEDs: a green LED to indicate normal functioning of the sensor according to and a red LED to indicate sensor fault. In case of the latter situation, through the NUCLEO-F411RE board, the valve will close, and a message indicating the necessity of changing or repairing the sensor will be sent.

IV. Experimental Results

To accomplish the implementation of the proposed approach, the architecture shown in Fig. 3 was applied. As shown in the figure, this is an elementary communication network composed of three main components: a computer, an embedded system, and a set of sensors and actuators.

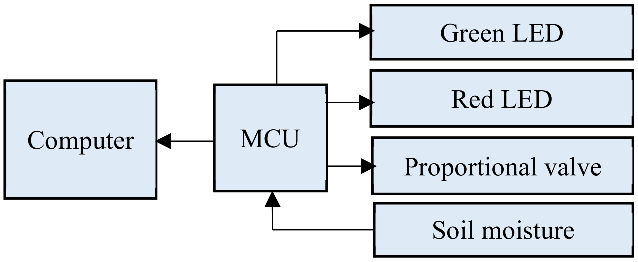

Fig. 3. System’s architecture.

The schematic of the testbed for the proposed approach is presented in Fig. 4, consisting of a 32-bit microcontroller embedded into a NUCLEO-F411RE board connected to a soil moisture sensor by the latter’s three pins: the analog output (AO), the ground (GND), and the voltage input (VCC). After setting pin A0 in the microcontroller as an analog input, D1 as the TX connectivity pin and D0 as the RX connectivity pin, sensor data were received through Algorithm 1 executed in a loop.

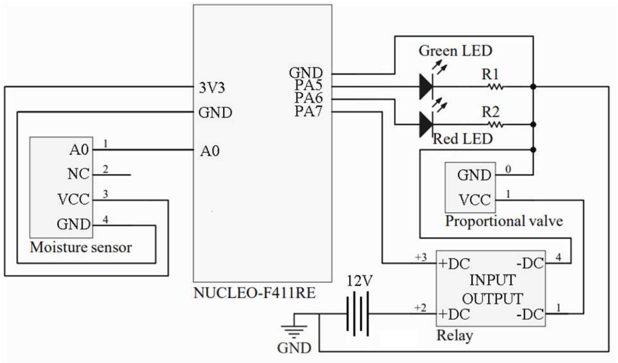

Fig. 4. Schematic diagram of the testbed.

```
Algorithm 1 Transmitting data through microcontroller
Input Raw analog values coming from sensor.
Output Soil moisture percentage.
for $k$ = 1 to 32 do
        Analog-to-digital conversion of input.
        Digital value is stored as part of an array.
end for
for $k$ = 1 to 32 do
        Print value stored in the array.
end for
```

Through Algorithm 1, data that enter the analog input are converted into digital values, which are stored in a 32-element array. Then, those 32 elements are printed from first to last. The printed values are received and plotted through Algorithm 2, executed through Python code.

```
Algorithm 2 Receiving data from microcontroller.
Input Printed digital value from 0 to 1023.
Output Plotted soil moisture values.
for $k$ = 1 to 32 do
        Convert digital value to soil moisture percentage.
end for
plot values
```

This code receives and splits the array sent by the microcontroller into 32 values, then each one of these quantities is multiplied by (100/1023) and subtracted from 100 to process each value in the range of 0 to 1023 into a moisture percentage from 0% to 100%, where 100% is the maximum moisture and 0% is the minimum moisture. Fig. 5 shows the plot created through Algorithm 2. Such plot shows normal moisture levels measured right after a moderate watering of the plant.

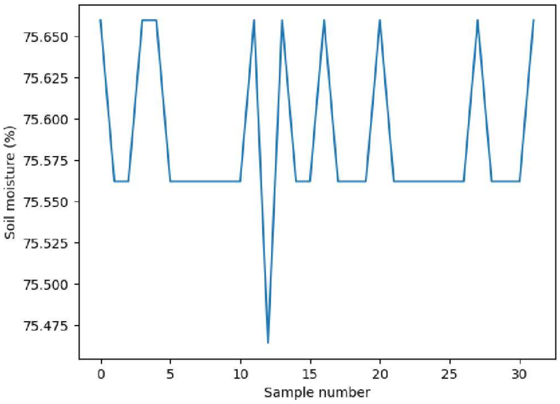

Fig. 5. Plot of 32 soil moisture percentages obtained from an array.

Additionally, and aiming to assess the behavior of the sensor, Euclidean distances of the Haar DWTs were calculated and plotted for 10 1D 8-element arrays. Algorithm 3 summarizes the procedure that allowed the obtention of these distances. This procedure generates a pattern 1D wavelet and a set of additional wavelets of the same dimension.

Algorithm 3 Calculation of Euclidean distances between DWTs
Input Soil moisture percentage
Output Euclidean distance between pattern and calculated wavelets

```
for $k$ = 1 to 8 do
    A soil moisture value is stored in an 8-element array.
end for
Approximation and detail coefficients are calculated for
  pattern wavelet.
for $k$ = 1 to 10 do
    for $k$ = 1 to 8 do
        A soil moisture value is stored in an 8-
        element array.
    end for
    Approximation and detail coefficients are calculated
    for wavelet.
    Euclidean distance between pattern and current
    wavelet is calculated.
    Euclidean distance is printed.
end for
```

Fig. 6 shows the plot of the 10 Euclidean distances obtained through Algorithm 3. As seen in this plot, all the values are within the range given *by* $\bar{x} \pm 3\sigma$ (red lines), also called control limits. This result indicates the sensor is properly measuring stable values of moisture, which is displayed with the activation of a green LED diode. However, if the results are outside these control limits, a red LED diode is activated, and the valve that waters the plant will close to allow the replacement or repairment of the sensor. These actions are summarized in Algorithm 4.

Algorithm 4 Sensor element activation
Input Euclidean distance between wavelets
Output Signal to activate led and modify valve closing

```
if $\bar{x} - 3\sigma <$ Euclidean distance $< \bar{x} + 3\sigma$
    turn green LED on
else if Euclidean distance $> \bar{x} + 3\sigma$ or
  Euclidean distance $> \bar{x} - 3\sigma$
    turn red LED on
    close valve
end if
```

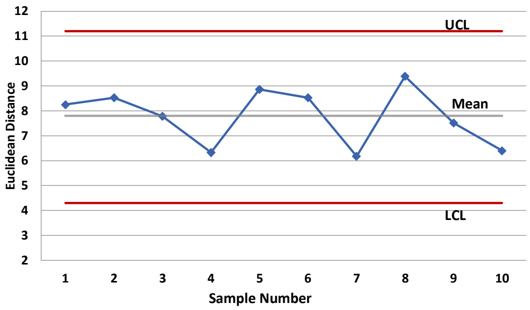

Fig. 6. Plot of 10 Euclidean distances.

The figure shows that all the values of these recent measurements are within the three-sigma limits from the mean, indicating a small distance between each value and the mean, hence a correct functioning of the sensor.

Once the normal behavior has been detected, the next step is to tamper with the sensor to verify if the method is capable of detecting the issue. Fig. 7 shows the comparison between normal and abnormal data. The abnormality was created by adding a variable resistance to the sensor and moving it around to simulate a variety of possible errors.

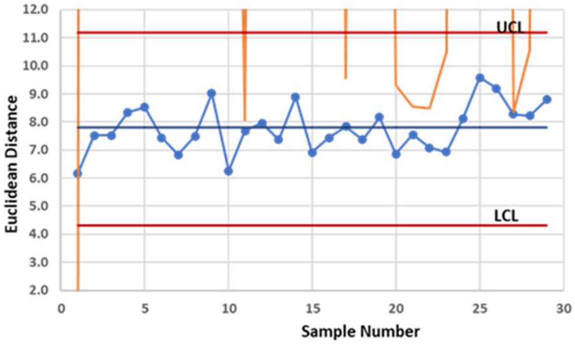

Fig. 7. Comparison between normal and abnormal behavior.

V. Conclusions and Future Work

As seen in this experience, it is possible to constantly track the behavior of a soil moisture sensor through the use of DWT and Euclidean distances. By monitoring the proper functioning of this device, it becomes possible to constantly cater to the moisture requirements of a plant as per the latest botany research, ultimately resulting in healthy growth.

In general, the implementation of the DWT in an embedded system can be highly beneficial in the monitoring of its sensory components. In broader scenarios such as large-scale cultivation and care of green areas, in which hundreds and even thousands of sensors are working simultaneously, this is a fast and economical alternative to make sure devices are working as expected.

Further research is the consideration of multiple variables in the sensing process, implying the activity of more sensors and the necessity of monitoring a broader scope of devices. In fact, besides soil moisture, light, nutrients, and temperature are highly relevant in the development of these living organisms, then sensors of these variables should receive a similar treatment. An area of future research is also the application of more complex forms of both discrete and continuous wavelet transform, which will probably provide more accurate information regarding the functioning of the sensors.

REFERENCES

[1] D. Li, Y. Wang, J. Wang, C. Wang, and Y. Duan, “Recent advances in sensor fault diagnosis: A review,” Sens. Actuators A: Phys., vol. 309. Elsevier B.V., July 2020.

[2] G. Troni and L. L. Whitcomb, “Field sensor bias calibration with angular-rate sensors: Theory and experimental evaluation with application to magnetometer calibration,” IEEE/ASME Trans. Mechatron., vol. 24, no. 4, pp. 1698–1710, August 2020.

[3] N. Jihani, M. N. Kabbaj, and M. Benbrahim, “Sensor fault detection and isolation for smart irrigation wireless sensor network based on parity space,” Int. J. Electr. Comput. Eng., vol. 13, no. 2, pp. 1463–1471, April 2023.

[4] X. Han, J. Jiang, A. Xu, A. Bari, C. Pei, and Y. Sun, “Sensor drift detection based on discrete wavelet transform and grey models,” IEEE Access, vol. 8, pp. 204389–204399, 2020.

[5] M. Pereira and B. Glisic, “Detection and quantification of temperature sensor drift using probabilistic neural networks,” Expert Syst. Appl., vol. 213, March 2023.

[6] T. Guo, T. Zhang, E. Lim, M. Lopez-Benitez, F. Ma, and L. Yu, “A review of wavelet analysis and its applications: Challenges and opportunities,” IEEE Access, vol. 10, pp. 58869–58903, 2022.

[7] M. Rhif, A. ben Abbes, I. Farah, B. Martínez, and Y. Sang, “Wavelet transform application for/in non-stationary time-series analysis: A review,” Appl. Sci., vol. 9, no. 7, p. 1345, March 2019.

[8] J. Pacheco, V. H. Benitez, L. C. Felix-Herran, and P. Satam, “Artificial neural networks-based intrusion detection system for Internet of Things fog nodes,” IEEE Access, vol. 8, pp. 73907–73918, 2020.

[9] J. Pacheco and S. Hariri, “Anomaly behavior analysis for IoT sensors,” Trans. Emerg. Telecommun. Technol., vol. 29, no. 4, April 2018.

[10] Q. Yaseen, F. AlBalas, Y. Jararweh, and M. Al-Ayyoub, “A fog computing based system for selective forwarding detection in mobile wireless sensor networks,” IEEE Int. Workshops on Found. and Appl. of Self* Syst. (FAS*W), pp. 256–262, December 2016.

[11] Y. Yang, L. Wu, and W. Hu, “Security architecture and key technologies for power cloud computing,” 2011 IEEE Int. Conf. on Transp., Mech., and Elect. Eng. (TMEE), pp. 1717–1720, 2011.

[12] O. Can and O.K Sahingoz, “A survey of intrusion detection systems in wireless sensor networks,” IEEE 2015 6th Int. Conf. on Model., Simul., and Appl. Optim. (ICMSAO), pp. 1–6, 2015.

[13] R. Godina and J. C. O. Matias, “Quality control in the context of industry 4.0,” Springer Proc. Math. Stat., 2019, vol. 281, pp. 177–187.

[14] D. C. Montgomery, Statistical Quality Control, 7th ed., Chichester: John Wiley & Sons, 2012.