

Hybrid Anomaly Detection Model on IoT Devices

Paul D. Rosero-Montalvo , Zsolt István , Pinar Tözün , and Wilmar Hernandez , *Senior Member, IEEE*

Abstract—Most machine learning (ML) proposals in the Internet of Things (IoT) space are designed and evaluated on pre-processed datasets, where the data acquisition and cleaning steps are often considered a black box. There are numerous challenges in IoT environments related to acquiring data from sensors. Therefore, the data acquisition stage requires additional data cleaning/anomaly techniques, which translate to additional resources, energy, and storage. We propose to carry out such techniques not in the cloud servers, where they would be usually done as part of an ML pipeline, but closer to the data source, on the IoT device itself. Among the benefits of this is the considerable reduction in the cost of implementation, due to less movement of data between IoT devices and the cloud. Consequently, we define three anomaly detection steps using smoothing filters, unsupervised learning, and deep learning techniques (hybrid model) to detect the different variations of anomalies, focusing on a small computational/memory footprint. This is essential for ensuring that the proposed model is feasible on the IoT device. The deployment of the final version of our hybrid model on AVR, Tensilica, and ARM microcontrollers showed that the last ones are an adequate target to implement such a model because they satisfied the necessary hardware requirements best. In short, the model consumes 50 KB of Flash and 12 KB of RAM to run the hybrid model locally, reducing the bandwidth by 60%.

Index Terms—outlier detection, data tampering, hybrid model, IoT.

I. INTRODUCTION

Nowadays, many research fields use machine learning (ML) techniques due to their capability to identify and describe patterns [1]. The datasets that form the basis of such techniques can be found in a variety of different formats such as: image, voice, and time-series data [2]. Often, the source of the dataset are sensors and similar devices “interacting” with the real world, sending their collected data to high-performance servers located in the cloud [3]. These servers are the ones used to run ML algorithms because such algorithms are computationally complex – naturally, a lot of effort in ML proposals goes into optimizing the computations in the cloud and making such servers run more efficiently. Nevertheless, many proposals that apply ML ideas on top of IoT/sensor data assume that the dataset has been “cleaned” and is typically treated as a black box [4].

This research was supported in part by Novo Nordisk Fonden (award ref: NNF20OC0064411), in part by the Corporacion Ecuatoriana para el Desarrollo de la Investigacion y la Academia (CEDIA), Ecuador (Research project: CEPRA XII-2018-13), and in part by the Universidad de Las Americas (UDLA), Quito, Ecuador (Research project: IEA.WHP.21.02).

Paul D. Rosero-Montalvo and Pinar Tözün are with the Computer Science Department at the IT University of Copenhagen, Denmark. (e-mail: pauro@itu.dk and pito@itu.dk)

Zsolt István is with the Computer Science Department of the Technical University of Darmstadt, Germany. (e-mail: zsolt.istvan@cs.tu-darmstadt.de)

Wilmar Hernandez is with Facultad de Ingenieria y Ciencias Aplicadas, Universidad de Las Americas, Quito, Ecuador. (e-mail: wilmar.hernandez@udla.edu.ec)

In the Internet of Things (IoT) space, the ability of electronic devices to work close to the end-users or interact directly with the environment has conceived a variety of emerging applications, such as: homes, offices, cities, agricultural farms, and industry [5]. Nevertheless, IoT devices can generate incorrect sensor readings because they are used in indoor and outdoor environments; often with low/no maintenance. Additionally, extreme weather conditions can impair performance and cause component wear [6]. Therefore, IoT devices periodically send collected data to the cloud for further processing. By some estimates, 80% of the battery consumption of IoT devices is related to wireless communication with the edge/cloud [7]. Moreover, in some scenarios, IoT devices send data to the cloud without bandwidth restriction/limitation carrying on bottlenecks and unnecessary storage on the server when data is not processed for a specific target [8].

Data collected are frequently inaccurate and sending them to the cloud can decrease the accuracy of the ML model. Besides, malicious users close to the IoT device can compromise the integrity of the information causing unpredictable outputs from the ML model [9]. Therefore, detecting anomalies must be done before sending data to the cloud. Anomaly detection techniques allow recognizing if data is statistically different from the rest, and such differences can be divided into intrinsic error (genuinely fault sensor), sensor event (an actual event that increases or decreases its value dramatically), and intermittent sensor error (data tampering) [10].

The most representative anomaly detection techniques fall into four categories: statistical, unsupervised, supervised, and deep learning. However, single models have not proven to have an exceptional performance [4], [11], [12]. Moreover, several works focused on deploying a single anomaly detection model show that such setups do not prevent the possibility of sending erroneous data to the cloud [5]. Therefore, in recent years, hybrid models are getting more attention and demonstrate prominent advantages when a set of anomaly detection techniques are merged into one single solution [7]. Nevertheless, a hybrid system with several anomaly detection algorithms is typically deployed outside the IoT device due to computational constraints, limited storage capacity, and battery power limitations [13].

Recent works [14], [15] deploy a decentralized solution where edge devices collaborate to train anomaly detection models. Besides, [12] presents two anomaly detection approaches running on the device. Lastly, [16], proposes a data edge verification mechanism based on blockchain to ensure that data is not tampered with. As illustrated by these works, a new trend exists to deploy a decentralized computational architecture where IoT devices are part of the ML pipeline running the inference of the model.

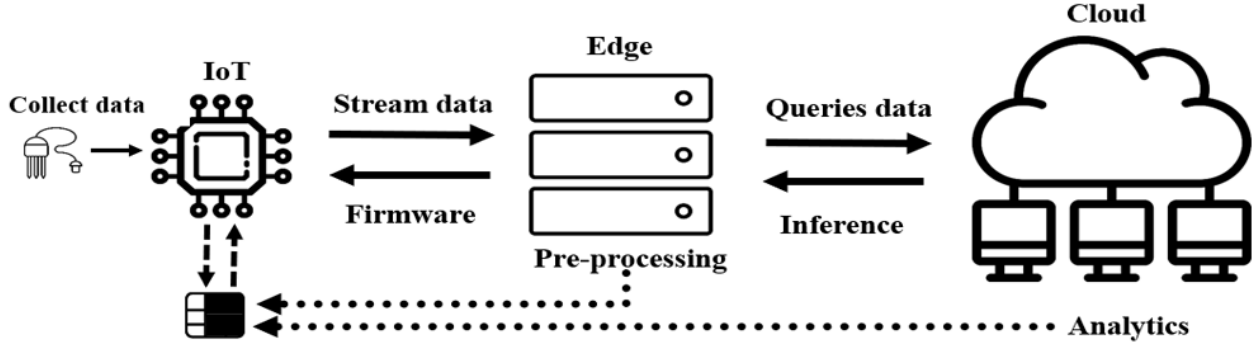


Fig. 1. Traditional data movement: solid black arrows, lightweight ML models exported to the IoT device from different locations: dotted arrows, and Hybrid model allocated in memory: square dotted arrows.

IoT: electronic devices with microcontrollers, Edge: Gateways with capabilities to deploy data pre-processing models such as payload errors, empty cells, and manning the wireless communication architecture; and Cloud: Powerful server with the ability to train heavy ML models

The goal of this paper is to reduce the cost of processing/detecting anomalies on the cloud and ensure the data integrity by processing data locally. Following novelty decentralized computational architectures [12], this work aims to achieve this goal by designing and deploying a hybrid ML model that detects anomalies on IoT devices. However, there are open challenges, such as finding both the suitable model for each anomaly detection technique and the optimum steps to deploy a hybrid anomaly detection model on the device. Related to the literature, the most representative anomaly detection techniques are as follows: noise reduction ([3], [17]), outlier detection ([18]–[20]), and tamper detection ([10], [21]). Based on [11], [14], we conduct an extensive literature review focused on determining the underlying algorithms and how these could be ported to emerging microcontrollers with better computational capacity. Therefore, we explore a range of possible algorithms to determine how well they work on small IoT devices.

To prove our proposal on how to develop the hybrid model locally, Figure 1 shows in solid black arrows how traditionally data is sent to the Edge where it is stored. Then, from the cloud side, computer scientists query data to develop machine learning models [22]. Next, the inference goes back to the Edge to update the firmware to send it back to the IoT device. In contrast, with the computational power of new microcontrollers, the dotted arrows show how some ML algorithms that typically work outside the device can be quantized to be processed locally in the IoT memory (shown in square-dotted arrows in Figure 1). As a result, our proposal aims to reduce the data movement, latency, and risk of compromising data sending them through communication networks. The contributions of this work can be summarized as follows:

- We propose a three-step hybrid anomaly detection model targeting to run on an IoT device considering the most relevant anomaly detection approaches (noise reduction, outlier detection, and tamper detection).
- We determine a set of anomaly detection algorithms from the literature review for each of the three steps mentioned above and evaluate them in terms of accuracy and metrics like memory consumption, and bandwidth on the IoT

device.

- Our results demonstrate that for the first, noise reduction, step smoothing filters, specifically the average and Gaussian algorithms, are the most effective to get improved samples. Then, the unsupervised learning methods are a suitable solution for the second, outlier detection, step. Especially, the one-class SVM algorithm reduces scatter significantly when it comes to obtain cleaned samples. Finally, for the third, tamper detection, step, a model based on deep neural networks can detect data tampered adequately if the algorithm is trained with the tamper possibilities. The final set of samples that pass through all these steps are the trusted ones and can be sent to the cloud.
- After determining the most suitable technique for each step, the hybrid anomaly detection model is tested in real environments with three microcontroller families, demonstrating that ARM microcontrollers are an adequate target to deploy this hybrid model satisfying the necessary hardware requirements.

The rest of the manuscript is structured as follows: Section II shows related works. Then, Section III presents the hybrid anomaly detection model proposed. Next, results are discussed in Section IV. Finally, Section VI presents the conclusions of the paper with a summary and directions for future work.

II. RELATED WORKS

There are several works with different approaches to detecting anomalies. Consequently, Works such as Gosh et al. [23], Gaddam et al. [24], Choi et al. [1] and Cook et al. [25] presented earlier state-of-the-art studies showing the most relevant techniques used for detecting anomalies. These works offer the statistical-based approaches, which assume some probabilistic data distribution categorized into parametric or non-parametric methods. In addition, they mention that the classification methods need prior knowledge about the data classes. Moreover, unsupervised learning is used when data is given but no corresponding output variables and deep learning with several types of neural networks can find light changes in the signal to detect anomalies.

TABLE I
OVERVIEW OF HYBRID ANOMALY DETECTION MODELS
TECHNIQUES: STATISTICAL (★), SUPERVISED (◆), UNSUPERVISED (■), AND DEEP LEARNING (▲)

Reviewed Works	Technique	Algorithms	Anomaly method	Model Location
Wu et al. [3]	◆ ▲	LSTM and Gaussian Naive Bayes	Noise reduction	Edge
Mansoor et al. [17]	◆ ■	SVM, k-NN, Bayes, Local outlier	Noise reduction	Edge
Sood et al. [10]	★ ◆	Regression trees, Random forest and SVM	Tamper detection and attacks	Cloud
Nesa et al. [18]	★	Regression Trees, Gradient Boosting Machine and Linear Discriminant Analysis	Outlier detection	Edge
Zhang et al. [4]	★ ◆	Filters, SVR, k-NN	Tamper/Noise detection	Server
Awawdeh et al. [19]	★ ■	Regressions model, iForest and Standard deviation	Outlier detection	Cloud
Zidi et al. [21]	◆	SVM and Naive Bayes	Tamper detection	Cloud
Deng et al. [20]	■	One class support machine	Outlier detection	Edge
Wang et al. [29]	■	Nearest-neighbor assembles and One-class SVM	Outlier detection	Cloud
Zhang et al. [30]	◆ ▲	Neural Network	Outlier/noise detection	Cloud
Zhang et al. [16]	★ ▲	Filters and Neural Network	Outlier/noise detection	Edge
Liu et al. [31]	▲	Convolutional neural network-long short-term memory	Outlier/noise detection	Edge
Alghanmi et al. [11]	◆ ■	Decision Tree and Hierarchical Affinity Propagation	Outlier detection	On-device
Garmaroodi et al. [12]	◆ ■	SVM and Hierarchical Affinity Propagation	Noise detection	On-device

Anomaly detection in IoT environments got attention since Zhang et al. [26] presented the first intuition about it and on how data with errors can impair the actuation of control systems. This work opens the opportunities to other researchers like Shen et al. [27] which presents the importance of the data validation stage before processing them. Later, Chacko et al. [28] presented the risks of bringing incorrect information to the cloud and how lightweight machine learning algorithms can be compiled in IoT environments. Besides, Yu et al. [5] presents a principal component analysis-based data outlier detection and sensor data. This work defines an architecture for sensor aggregation in the cloud. Then, Nesa et al. [18] considers different techniques to reach outlier detection. As a result, this work presented metrics to compare those algorithms with classification and statistical approaches to determine the performance of each technique. Wang et al. [29] used several algorithms such as isolation forest and local outlier factor; and their mathematical approach to present their isolation nearest neighbor ensemble (iNNE) technique. Then, they use different datasets to demonstrate the accuracy of their model, which is over 90%. Conversely, Sood et al. [10] presents the first initiative to define a high-level tamper detection combining some algorithms with different approaches: statistical, supervised, and unsupervised techniques. Therefore, the model complexity increases at the same time as its robustness. An external dataset has been used in this work to evaluate the model's performance.

Recently, anomaly detection has been studied based on new trends when the flow of information processed by the IoT data has grown exponentially. For these trends, Zhu et al. [7] propose a novel framework named GAAOD (Grid-based Approximate Average Outlier Detection) to support KNN-Based

outlier detection over IoT streaming data. Furthermore, Zhang et al. [4] propose a three-level hybrid model based on: median filter (MF), empirical mode decomposition (EMD), classification and regression tree (CART), autoregression (AR), and exponential weighted moving average (EWMA) methods (called MF-EMD-CARTAR-EWMA) to detect outliers in sensor data. The hybrid model was evaluated using a real hydrometeorological observation network dataset. Then, Wu et al. [3] define a long short-term memory (LSTM)-Gauss-Bayes method, which is a synergy of the long short-term memory neural network (LSTM-NN) and the Gaussian Bayes model for outlier detection in the IoT devices. Next, the new tendency to use decentralized computational systems, Garmaroodi et al. [12] present a smarter IoT system applied in pharmaceutical systems to detect anomalies in water purifying pumps. They process data locally and send relevant information to the edge to create sensor fault reports. In addition, Liu et al. [31] and Wang et al. [15] demonstrate an application of federated learning into anomaly detection in industrial environments using hybrid models.

Summarizing the most relevant works and their approach, Table I shows the applied algorithm, the used algorithm, and the location where the ML algorithm was developed. Also, Table I shows relevant information about the algorithm selection for the following sections presented in this research. However, even when previous works presented several solutions to face anomaly detection in sensor data and IoT devices, there are open challenges. For example, bringing these solutions into the IoT device due to the new computational capabilities of the microcontrollers. Moreover, several works present anomaly detection hybrid models using different ML methods. Nevertheless, tamper detection techniques are not

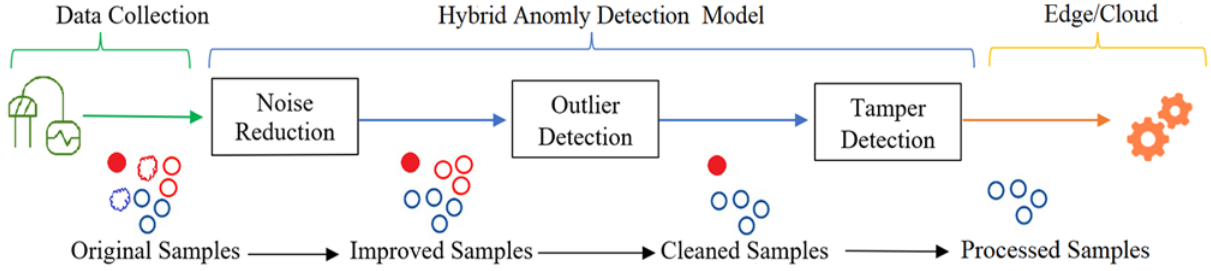


Fig. 2. Hybrid Anomaly Detection Model (→), dataflow (→), errors , outliers , tampered data 

deployed as part of hybrid models, even with the increasing concerns about data confidentiality.

III. HYBRID ANOMALY DETECTION MODEL

ML applications in IoT environments start with the data collection by sensors, the next step is data anomaly detection. However, even though usually just the model inference step is deployed on the IoT device [32], only some techniques can be exported to IoT devices due to their expensive hardware requirements. In addition, these anomaly detection techniques are applied in different ways to reach individual purposes, such as: noise reduction (improved samples), outlier detection (cleaned samples), and tamper detection (processed samples). Therefore, it is necessary to present a hybrid model merging the mentioned methods to detect erroneous, unpredictable, and unexpected data [33]. However, this hybrid model must be compiled with the minimum computational footprint [34]. As a result, the IoT device stores in memory and runs in sequence different ML models related to detecting all the anomaly possibilities. Figure 2 shows the following steps of the proposed hybrid model and the corresponding data flow, starting with the original samples, which have noise (represented by variations), outliers (represented by red circles), and tampered data (represented by red dots).

A. Data collection

We propose a controlled data acquisition environment where data manipulation is doable and the hybrid model can be tested. Therefore, we employ a well-known CO₂, temperature, and humidity sensor SCD-30, which is applied in several indoor and outdoor system applications [35]. First, the electronic system has one microcontroller, the SCD30 sensor, and battery, which receives data in normal conditions in an office where people work for around 7 hours in groups of two or three people (Label 1). Second, people try to tamper with the sensor by blowing on it (Label 2). Third, people smoke near the sensor to activate the fire alarm (Label 3). Finally, we simulate a fire by putting fireballs in a basket near the sensor (Label 4).

However, because the data needs a couple of hours to get tiny variations and the security of the building can be affected if malicious users try to tamper with the sensor, we create a test box (scaled size of a standard office) to accelerate the data acquisition process. Figure 3 shows the design with a computer fan to cool the box, an incandescent bulb to warm up the box, and a small door that the user can tamper the

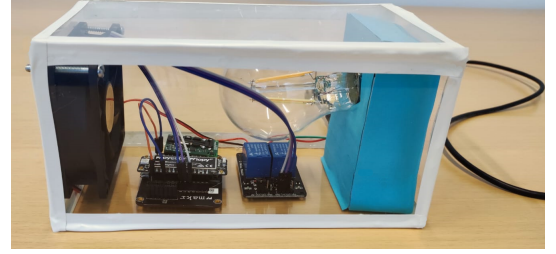


Fig. 3. Test-box designed for data acquisition process. (1) Computer fan, (2) Light bulb, and (3) electronic circuit

sensor directly. The sample rate for label 1 in offices was 10 minutes. Then, the trend was calculated to define the same seasonality behavior in the text box, which is to take samples every 5 seconds. As a result, the data was stored into a matrix $X \in \mathbb{R}^{m \times n}$, where m is the number of samples and n amounts the number of variables (that is to say, the number of attributes representing each sample). Meanwhile, $L \in \mathbb{R}^{m \times 1}$ is a vector holding the sample labeling. In this case, $m=1500$ and $n = 3$.

B. Noise reduction

A phenomenon can be described by static data, time series, and images. Therefore, sensors measure the magnitude of a quantity and convert it into an electric signal at a specific sample rate (original samples) [4]. Then, the signal is discretized/digitalized to be understandable to the microcontroller. However, errors like voltage fluctuations, non-linearity response, and vibrations insert noise into the electric signal, confusing the ML algorithm in the feature extraction stage. For this reason, signal smoothing is a filter that reduces these noise components when the phenomenon does not have high sampling frequencies getting a cleaner signal [36]. Consequently, Moving Average, Mean, Gaussian filter, and Savi-Golay are the most relevant smoothing algorithms [37]. In this hybrid model, the output of this step is called *improved samples*.

C. Outlier Detection

This stage is responsible for detecting data with a different distribution than the rest. This process is carried out through an unsupervised analysis [4]. As a result, a refined data subset called *cleaned samples* is obtained. Outlier detection methods used for this stage related to literature review are as follows:

- **The standard deviation method (STD)** quantifies the variation or spread of a numerical data set in which two

standard deviations represent 95% of the data. Therefore, any data point outside of this boundary is removed [17].

- **The Local Outlier Factor (LOCAL)** measures the local deviation of the density of a given sample concerning its neighbors. It is local in that the anomaly score depends on how isolated the object is from the surrounding neighborhood [19].
- **Isolation Forest (ISO)** is based on modeling the expected data in such a way as to isolate anomalies that are both few and different in the feature space [18].
- **Elliptic Envelope (ELLI)** detects outliers in a Gaussian distributed dataset. It provides the *contamination* argument that defines the expected ratio of outliers observed in practice [19].
- **One-Class SVM (OSVM)** captures the density of the majority class and classifies examples on the extremes of the density function as an outlier [20].

D. Tamper Detection

In some scenarios, the information acquired by the IoT device can be compromised by malicious users trying to steal or modify data. For example, when the ML algorithm creates the model, tampered data can affect the rules creation, and outputs cannot be what the ML developer expected [22]. In contrast, the IoT device may have the proper model, whereas users try to insert erroneous inputs creating manipulated environments (e.g., activate the fire alarm by putting a lighter near the smoke sensor) [38]. Due to the above-mentioned statements, the tamper-detection stage requires knowing all the manipulation possibilities to detect them, which needs to deploy supervised learning. This last step produces *processed samples* to be sent to the cloud. The algorithms used to reach this goal are presented below:

- **k-Nearest Neighborhood** associates a new instance with the training base and assigns it to the closest group according to its distance [4].
- **Naive Bayes** describes the probability of an event happening based on previous knowledge of the event [21].
- **Support Vector Machine (SVM)**: has kernel functions that put the data in hyperplanes to improve the decision boundaries [17].
- **Decision Tree**: is a heuristic model based on rules [11].
- **Neural Network**: is a recursive training model [2].

IV. RESULTS

This section evaluates the choice of algorithms listed for each step in Section III, and determines a target microcontroller family to deploy the final version of the proposed hybrid model.

A. Experimental setup

ML models are trained on an external computer (core i7 with 32 GB of RAM) with the previously obtained data set, called $X \in \mathbb{R}^{m \times n}$. Therefore, regression analysis is used to observe how smoothing algorithms soft the signal in the noise reduction step, which improves samples (according to the data

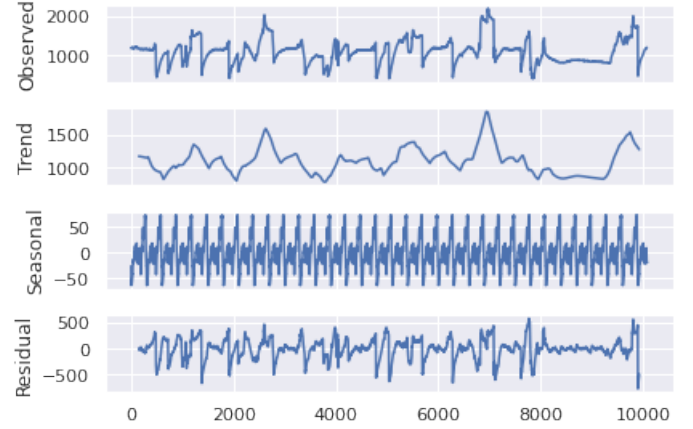


Fig. 4. Main components of CO₂ signal. Y-axis: Analog digital conversion values (0-1023), X-axis: number of samples

flow presented in Figure 2). Then, the ANOVA statistical metric determines the improved variability of the samples and how unsupervised models reduce the scatter, obtaining cleaned data. Next, the classification performance determines which classification method is adequate for detecting tampering in the cleaned data to send to the cloud just processed data. Finally, selected algorithms are exported as a .h file (i.e., header file referenced by a document written in C) to determine the memory consumption. It is important to mention that the original samples were divided 10 times randomly to avoid bias in selected algorithms.

B. Improved samples

The data gathering process is affected by noise injection from the abovementioned sources. Therefore, time series analysis provides many techniques to understand a dataset better. Perhaps, the most useful of them are Level (i.e., the base value if it were in a straight line), Trend (i.e., the linear increasing or decreasing behavior), Seasonality (i.e., the repeating patterns), and Noise (i.e., the variability of the observations). In this way, Figure 4 shows these components from the CO₂ variable as an example. It also proves that the CO₂ signal contains noise. Therefore, smoothing filters are required, where noise samples cancel each other out.

The above-mentioned smoothing filters are deployed to reduce the dataset noise and get improved samples. Hence, regression analysis such as the Root Mean Square Error (RMSE) determines how close the observed data points are to the filter-smoothed values. In addition, the R2 metric presents the fitness of the filter, and the signal-to-noise ratio (SNR) shows improvement in the quality of the relevant signal. Therefore, the metrics mentioned above were used to select a suitable smoothing filter. As a result, even though the Median filter has a better RMSE metric for each signal (i.e., minor value), its output is very similar to the signal input and does not reduce noise adequately. Conversely, for the CO₂ and Temperature signal, the Moving average filter has both better SNR and R2 metric. Nevertheless, the Gaussian filter is selected for the humidity signal due to scoring higher in the

evaluated metrics. Figure 5 shows graphically how the moving average filter refines the CO₂, and Table II summarizes the values obtained from each metric where k stands for windows size.

TABLE II
SMOOTH FILTER STATISTICAL ANALYSIS

	Average k=30	Median k=30	Savi-Golay sigma=7	Gaussian k=9, poly=4	Statistic Metrics
CO ₂	6.67	5.61	5.87	6.41	RMSE
	0.99	0.97	0.99	0.96	R2 score
	8.40	7.99	7.97	8.26	SNR
TEMP	0.73	0.04	0.07	0.156	RMSE
	0.98	0.99	0.99	0.99	R2 score
	15.37	14.78	14.74	15.17	SNR
HUM	1.74	0.14	0.23	0.41	RMSE
	0.90	0.97	0.97	0.98	R2 score
	8.72	8.65	8.65	8.75	SNR

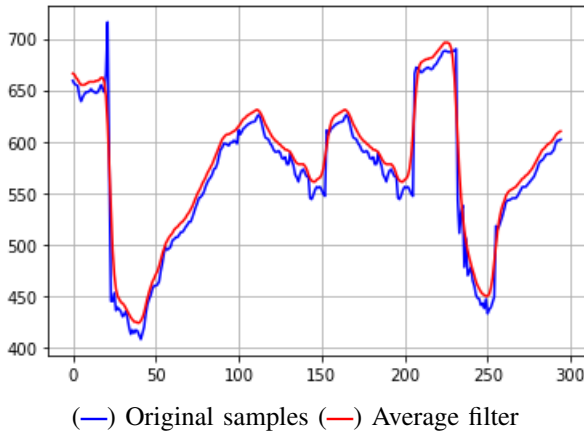


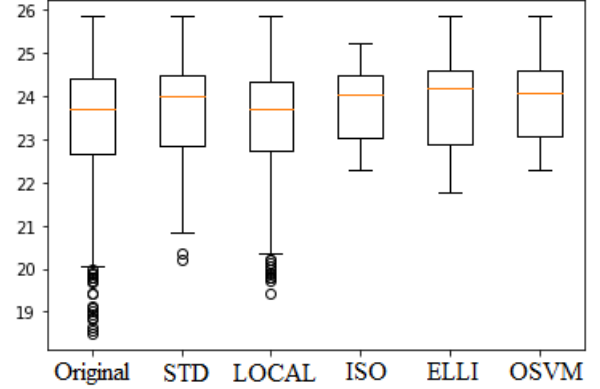
Fig. 5. Smoothing filter applied in CO₂ signal. Y-axis: Analog digital conversion values (0-1023), X-axis: number of samples

C. Clean Samples

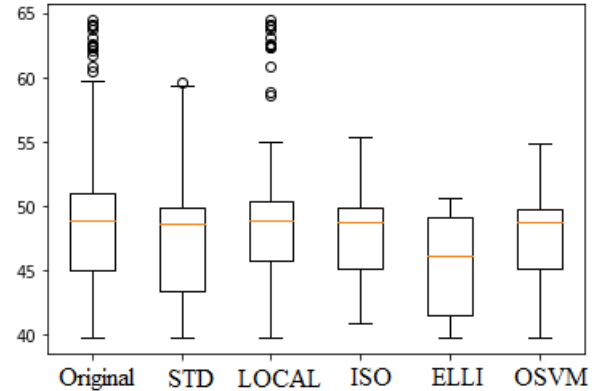
The dataset maintains outliers with different distributions. Therefore, statistical methods can describe the variability of the entire dataset, and ANOVA analysis can be used for this purpose. This analysis splits the dataset into systematic factors (i.e., data with normal distribution) and random factors (i.e., outliers). Then, unsupervised machine learning methods can prune these random factors and increase the decision edges distance between labels. Indeed, these outliers detection methods can reduce the data complexity, and the ML model can reach high throughput.

The mentioned-above outlier detection approaches are applied to the dataset to get different improved/pruned subsets (all methods prune data around 15%) and test them with ANOVA analysis. Consequently, the standard deviation and local Outlier Factor algorithms keep the data set scattered, which can cause confusing decision edges between labels. On the other hand, the Isolation Forest algorithm concentrates the data due to the central point (mean). Furthermore, Elliptic Envelope and One-Class SVM have similar results in maintaining data characteristics while reducing scatter. For this

reason, we selected the standard deviation, Isolation Forest, and One-Class SVM criteria, representing improvements in reducing data scatter to determine in the next step which of them better eliminates outliers improving the classification algorithm performance. Finally, Figure 6 shows the variability of each method obtained by ANOVA analysis.



(a) Temperature



(b) Humidity

Fig. 6. Variance data distribution by ANOVA analysis. Y-axes: Degrades (a) and Humidity percent (b)

D. Processed Samples

This stage aims to detect whether the sensor has been tampered with. Therefore, it is necessary to detect labels 2 (people blowing on the sensor) and 3 (people smoke near the sensor to active fire alarm). Hence, classification algorithms were trained with clean subsets (obtained in the previous section) and the original samples to determine improvements in the classification decision boundaries.

Table III shows the classification performance of each applied algorithm. As a result, the original samples demonstrates outliers because ML models have issues classifying incoming data correctly. In addition, despite the subset obtained by the Standard Deviation, reducing outliers has similar classification results to the original samples dataset. In contrast, ML models have high classification performance when they are trained with datasets obtained by Isolation Forest and One-class SVM algorithms. Hence,

TABLE III
CLASSIFICATION PERFORMANCE FOR EACH ANOMALY DETECTION
TECHNIQUE

ML model	Datasets			
	Orig. Samples	STD	ISO	OSVM
SVM	0.59	0.70	0.61	0.75
k-NN	0.83	0.89	0.90	0.93
Naive Bayes	0.72	0.80	0.80	0.81
Decision Tree	0.88	0.90	0.96	0.95
Neural Network	0.94	0.93	0.98	0.98

One-class SVM is adequate to be implemented in the IoT device due to its capacity to reduce outliers significantly and lightweight memory footprint.

Nevertheless, neural networks can achieve close to 100% accuracy in all the evaluated scenarios. However, its complexity (i.e., the number of layers and neurons) is considerably reduced. For example, the optimal neural network should have two hidden layers of 8 and 4 neurons with the original database, respectively. However, with the database obtained by One-class SVM, only a hidden layer of 6 neurons is necessary, which significantly improves memory consumption.

Once the ML models are trained, the decision tree algorithm and neural network show higher classification results. However, in real test scenarios, the decision tree algorithm has strict rules which reduce the ability to detect light changes in incoming data. Therefore, the neural network model is an adequate solution to detect whether the sensor has been tampered with.

E. Memory Consumption

One essential part of a microcontroller is its memory, which stores information temporarily or permanently. Hence, memory blocks are described as arrays and divided into cells that can store data and be accessed using a unique identifier, representing its address or position relative to the memory array. Consequently, memory blocks can be either volatile or non-volatile, and the control processing unit “CPU” accesses them concerning the computer architecture; This can be either Von Neumann or Harvard architecture [39]. In this scenario, the proposed hybrid model is tested in AVR, Tensilica, and ARM microcontrollers. While AVR and Tensilica family microcontrollers are based on the Harvard architecture model, ARM family microcontrollers can be based on either von Neuman or Harvard architectures. These microcontroller families divide memory units into two. First, is the Flash memory, in which the firmware is stored to be executed. Second, is the Random Access Memory (RAM), where microcontrollers typically use SRAM memory, which is a type of RAM that uses flip-flops to store one bit of data.

The electronic system deployed uses an Arduino Nano from the **AVR family** with 32kB in Flash and 2kB in RAM. An Arduino Nano 33 IoT from **ARM family** with 256kB in Flash and 32kB in RAM, and NodeMCU from **Tensilica family** with 4MB in Flash and 128 kB in RAM. Each level of the hybrid model was compiled and tested to determine the memory consumption.

TABLE IV
FLASH CONSUMPTION ANALYSIS

Block memory	Microcontrollers		
	AVR	ARM	Tensilica
Boot-loader	450 B	11 kB	260 kB
Main program	12 kB	14 kB	20 kB
Noise reduction	1kB	2 kB	2 kB
Outlier detection	15 kB	20 kB	20 kB
Tamper detection	5 kB	7 kB	7 kB
Total	≈ 33 kB	≈ 56 kB	≈ 310 kB
Free memory	≈ 0	≈ 200 kB	≈ 4500 kB

TABLE V
RAM CONSUMPTION ANALYSIS

Block memory	Microcontrollers		
	AVR	ARM	Tensilica
Boot-loader	10 B	3 kB	27 kB
Main program	400 B	3 kB	3 kB
Noise reduction	500B	1 kB	1 kB
Outlier detection	500 B	2 kB	2 kB
Tamper detection	1.5 kB	3 kB	3 kB
Total	≈ 3 kB	≈ 12 kB	≈ 37 kB
Free memory	≈ 0	≈ 20 kB	≈ 25 kB

Tables IV and V summarize the memory consumption of each microcontroller. As relevant results, AVR microcontrollers do not have enough memory to compile this application because the Flash needed is 35 kB, and SRAM is fully utilized (2kB). Besides, the outlier detection model uses half of the RAM. ARM microcontrollers, specifically with microcontrollers M0, M4, and recently M7, are designed to allow ML models with higher computational resources than the AVR family. Arduino Nano IoT uses its Flash around 11 kB in the bootloader, 12 kB in the main program, 2 kB to allocate the smoothing filter, 20 kB for the outlier detection model, and 5 kB for the tamper detection. Furthermore, Flash and SRAM utilization were 56 kB and 12 kB, respectively. Therefore, Nano IoT has at least 200 kB free memory, and the 20 kB RAM is available. On the other hand, since Tensilica microcontrollers need external libraries to be programmed in the Arduino environment, the bootloader needs 260 kB, which is a high memory requirement. Hence, the Flash memory was used at 310 kB for the hybrid model; it is around 13 % of the total Flash memory. Nevertheless, these boards struggle with some WiFi libraries that are inefficiently compiled.

F. Bandwidth Consumption

To determine the impact of performing the anomaly detection on the IoT device instead of the cloud on bandwidth consumption, we calculate the expected bandwidth consumption for each case. First, sending streaming data to the cloud, the payload uses 20 bytes, and the data packet (i.e., control header) is 30 bytes sending data to the cloud each 30 ms by MQTT protocol, which means the bandwidth is around 1 Kbps. Second, we define sending data to the cloud when the system detects a tamper situation since the outlier detection can be processed locally by its outputs (fan). Hence, the same data packet and protocol use just 300 bps (worst scenario) since the frequency of sending messages decreases significantly. Besides, we determine that outliers appear when the IoT device

wakes up from sleep modes, even when the main program waits until the sensor responds that is calibrated. Therefore, this anomalous behavior is corrected locally. In addition, data is not compromised because packages sent to the cloud can be just alert messages reducing, even more, the bandwidth needed.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, the proposed three-step anomaly detection hybrid model, which integrates data pre-processing (reducing errors), and outlier detection tasks, achieves excellent performance in the data tamper detection scenarios for data collection from IoT devices. Besides, the hybrid model runs on IoT devices to process data locally with a minimal memory footprint. In the first step, to achieve noise reduction, smoothing algorithms demonstrate a suitable solution for canceling voltage fluctuations produced by sensors. In the second step, we demonstrate that it is necessary to implement an outlier detection step to increase the robustness of detecting abnormal distribution in data. Therefore, One-Class SVM prunes outliers and reduces the scatter better. Lastly, in the third step, the tamper detection step needs previous knowledge to train models and detect anomalous sensor behavior. Thus, neural networks can recognize those changes in sensor functionalities and detect whether if the work environment is compromised. The anomaly detection hybrid model also proves the new functionalities that emerging microcontrollers may have processing ML models close to the user. This reduces latency with fewer data movements between devices with adequate time response. In addition, processing data locally reduce the opportunity for malicious users to steal data. Furthermore, neural networks can recognize 98% of tamper situations.

This research provides a new perspective on anomaly detection. The proposed hybrid model was evaluated in time series data from one sensor, which can collect three different variables (i.e., CO₂, temperature, and humidity). In future work, the proposed method will be further expanded and optimized for different sensor data with extensive power consumption analysis.

REFERENCES

- [1] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines," *IEEE Access*, vol. 9, pp. 120 043–120 065, 2021.
- [2] Y. Djenouri, D. Djenouri, A. Belhadi, G. Srivastava, and J. C.-W. Lin, "Emergent deep learning for anomaly detection in internet of everything," *IEEE Internet of Things Journal*, pp. 1–1, 2021.
- [3] D. Wu, Z. Jiang, X. Xie, X. Wei, W. Yu, and R. Li, "Lstm learning with bayesian and gaussian processing for anomaly detection in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5244–5253, 2020.
- [4] M. Zhang, X. Li, and L. Wang, "An adaptive outlier detection and processing approach towards time series sensor data," *IEEE Access*, vol. 7, pp. 175 192–175 212, 2019.
- [5] T. Yu, X. Wang, and A. Shami, "Recursive principal component analysis-based data outlier detection and sensor data aggregation in iot systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2207–2216, 2017.
- [6] R. Kaur and J. Kaur Sandhu, "A study on security attacks in wireless sensor network," in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2021, pp. 850–855.
- [7] R. Zhu, X. Ji, D. Yu, Z. Tan, L. Zhao, J. Li, and X. Xia, "Knn-based approximate outlier detection algorithm over iot streaming data," *IEEE Access*, vol. 8, pp. 42 749–42 759, 2020.
- [8] A. Singh, P. Vepakomma, O. Gupta, and R. Raskar, "Detailed comparison of communication efficiency of split learning and federated learning," *CoRR*, vol. abs/1909.09145, 2019. [Online]. Available: <http://arxiv.org/abs/1909.09145>
- [9] A. K. Pathak, S. Saguna, K. Mitra, and C. Åhlund, "Anomaly detection using machine learning to discover sensor tampering in iot systems," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.
- [10] K. Sood, M. R. Nosouhi, N. Kumar, A. Gaddam, B. Feng, and S. Yu, "Accurate detection of iot sensor behaviors in legitimate, faulty and compromised scenarios," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2021.
- [11] N. Alghanmi, R. Alotaibi, and S. M. Buhari, "Hlmc: A hybrid learning anomaly detection model for unlabeled data in internet of things," *IEEE Access*, vol. 7, pp. 179 492–179 504, 2019.
- [12] M. S. S. Garmaroodi, F. Farivar, M. S. Haghighi, M. A. Shoorehdeli, and A. Jolfaei, "Detection of anomalies in industrial iot systems by data mining: Study of christ osmotron water purification system," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 280–10 287, 2021.
- [13] M. Keerthika and D. Shanmugapriya, "Wireless sensor networks: Active and passive attacks - vulnerabilities and countermeasures," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 362–367, 2021, international Conference on Computing System and its Applications (ICCSA- 2021). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666285X2100073X>
- [14] Y. Liu, L. Dang, S. Li, K. Cai, and X. Zuo, "Research progress on models, algorithms, and systems for remote sensing spatial-temporal big data processing," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 5918–5931, 2021.
- [15] X. Wang, S. Garg, H. Lin, J. Hu, G. Kaddoum, M. Jalil Piran, and M. S. Hossain, "Toward accurate anomaly detection in industrial internet of things using hierarchical federated learning," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7110–7119, 2022.
- [16] W. Zhang, J. Wang, G. Han, S. Huang, Y. Feng, and L. Shu, "A data set accuracy weighted random forest algorithm for iot fault detection based on edge computing and blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2354–2363, 2021.
- [17] M. A. Bhatti, R. Riaz, S. S. Rizvi, S. Shokat, F. Riaz, and S. J. Kwon, "Outlier detection in indoor localization and internet of things (iot) using machine learning," *Journal of Communications and Networks*, vol. 22, no. 3, pp. 236–243, 2020.
- [18] N. Nesa, T. Ghosh, and I. Banerjee, "Outlier detection in sensed data using statistical learning models for iot," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6.
- [19] M. Awawdeh, T. Faisal, A. Bashir, and A. Sheikh, "Application of outlier detection using re-weighted least squares and r-squared for iot extracted data," in *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, 2019, pp. 1–6.
- [20] X. Deng, P. Jiang, X. Peng, and C. Mi, "An intelligent outlier detection method with one class support tucker machine and genetic algorithm toward big sensor data in internet of things," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4672–4683, 2019.
- [21] S. Zidi, T. Moulahi, and B. Alaya, "Fault detection in wireless sensor networks through svm classifier," *IEEE Sensors Journal*, vol. 18, no. 1, pp. 340–347, 2018.
- [22] M. Shafique, T. Theocharides, V. J. Reddy, and B. Murmann, "Tinyml: Current progress, research challenges, and future roadmap," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, 2021, pp. 1303–1306.
- [23] N. Ghosh, K. Maity, R. Paul, and S. Maity, "Outlier detection in sensor data using machine learning techniques for iot framework and wireless sensor networks: A brief study," in *2019 International Conference on Applied Machine Learning (ICAML)*, 2019, pp. 187–190.
- [24] A. Gaddam, T. Wilkin, M. Angelova, and J. Gaddam, "Detecting sensor faults, anomalies and outliers in the internet of things: A survey on the challenges and solutions," *Electronics*, vol. 9, no. 3, 2020. [Online]. Available: [RAhttps://www.mdpi.com/2079-9292/9/3/511](https://www.mdpi.com/2079-9292/9/3/511)
- [25] A. A. Cook, G. Misirlı, and Z. Fan, "Anomaly detection for iot time-series data: A survey," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6481–6494, 2020.
- [26] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 12, no. 2, pp. 159–170, 2010.

- [27] Z. Shen and Q. Wang, "Data validation and confidence of self-validating multifunctional sensor," in *SENSORS, 2012 IEEE*, 2012, pp. 1–4.
- [28] V. Chacko and V. Bharati, "Data validation and sensor life prediction layer on cloud for iot," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2017, pp. 906–909.
- [29] Z.-M. Wang, G.-H. Song, and C. Gao, "An isolation-based distributed outlier detection framework using nearest neighbor ensembles for wireless sensor networks," *IEEE Access*, vol. 7, pp. 96 319–96 333, 2019.
- [30] K. Zhang, K. Yang, S. Li, D. Jing, and H.-B. Chen, "Ann-based outlier detection for wireless sensor networks in smart buildings," *IEEE Access*, vol. 7, pp. 95 987–95 997, 2019.
- [31] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6348–6358, 2021.
- [32] J. Park, S. Wang, A. Elgabli, S. Oh, E. Jeong, H. Cha, H. Kim, S. Kim, and M. Bennis, "Distilling on-device intelligence at the network edge," *CoRR*, vol. abs/1908.05895, 2019. [Online]. Available: <http://arxiv.org/abs/1908.05895>
- [33] L. Ravaglia, M. Rusci, D. Nadalini, A. Capotondi, F. Conti, L. Benini, and L. Benini, "A tinyml platform for on-device continual learning with quantized latent replays," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, pp. 1–1, 2021.
- [34] Y. Li, X. Wang, X. Gan, H. Jin, L. Fu, and X. Wang, "Learning-aided computation offloading for trusted collaborative mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 12, pp. 2833–2849, 2020.
- [35] Sensirion. Scd30 sensor. [Online]. Available: <https://sensirion.com/products/catalog/SCD30/>
- [36] J. Jeong, "Real-time whitening application to two microphone sensors for comb filtering and smoothing," in *2015 IEEE SENSORS*, 2015, pp. 1–4.
- [37] P. Kowalski and R. Smyk, "Review and comparison of smoothing algorithms for one-dimensional data noise reduction," in *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 2018, pp. 277–281.
- [38] J. Shafi and A. Waheed, "K-means clustering analysing abrupt changes in air quality," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2020, pp. 26–30.
- [39] A. E. Giakoumis, C. K. Volos, I. N. Stouboulos, H. M. Polatoglou, and I. M. Kyprianidis, "Chaos generator device based on a 32 bit microcontroller embedded system," in *2018 7th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2018, pp. 1–4.



Paul D. Rosero-Montalvo He is a Post. Doc at IT University of Copenhagen (ITU) in Denmark since June 2021. His research focuses on emerging microcontrollers to run machine learning models in decentralized networks. Earlier, He was a research assistant professor in the Applied Science Department at the Universidad Tecnica del Norte (UTN), Ecuador, for seven years. At the same time, he was a part-time lecturer in the TI Department at Instituto Tecnológico 17 de Julio, Ecuador. He received his Ph.D. from the University of Salamanca in Spain

in November 2020, where Prof. Vivian Lopez-Batista advised him. He has a master's degree in Data Management systems from Universidad de las Fuerzas Armadas ESPE, Ecuador (2018), and an Engineering degree in Electronics and Telecommunications from UTN (2013).



Zsolt István is a Full Professor at the Computer Science Department of the Technical University of Darmstadt, Germany, where he leads the Distributed and Networked Systems group. Earlier, he held positions at the IT University of Copenhagen, Denmark, and at the IMDEA Software Institute in Madrid, Spain. He earned his PhD title from the Systems Group at ETH Zürich, Switzerland.



Pinar Tözün is an Associate Professor at IT University of Copenhagen (ITU) in Denmark since 2018. Her research focuses on hardware-conscious machine learning, performance characterization of data-intensive systems, and scalability and efficiency of data-intensive systems on modern hardware. Before joining ITU, she was a research staff member at IBM Almaden Research Center (San Jose, CA, USA) for three years, where she was part of the team that led the development of IBM Db2 Event Store.

Before joining IBM, she received her PhD from École polytechnique fédérale de Lausanne (EPFL) in Switzerland in 2014. Her PhD thesis received ACM SIGMOD Jim Gray Doctoral Dissertation Award Honorable Mention in 2016.



Wilmar Hernandez received the degree in electronics engineering and the specialist degree in microelectronics from the Instituto Superior Politecnico Jose Antonio Echeverria (ISPJAE), Havana, Cuba, in 1992 and 1994, respectively, and the M.S. degree in signal treatment and the Ph.D. degree in electronic engineering from Enginyeria La Salle, Universitat Ramon Llull, Barcelona, Spain, in 1997 and 1999, respectively. From 1992 to 1995, he was a Lecturer with the Electrical Engineering Faculty, ISPJAE, and a Researcher with the Microelectronics Research

Center, ISPJAE. From 1999 to 2003, he was with the Department of Electronics and Instrumentation, University Institute for Automobile Research, Universidad Politecnica de Madrid (UPM), Spain, where he was the Technical Director of the department, from January 2003 to January 2004. From January 2004 to March 2013, he was an Associate Professor of circuits and systems with the Department of Circuits and Systems, EUIT de Telecomunicacion, UPM. From September 2014 to September 2015, he was a Researcher with SENESCYT, Ecuador, under the Prometeo Fellowship Program. From December 2015 to November 2017, he was a Professor with the Universidad Tecnica Particular de Loja, Ecuador. Since January 2018, he has been a Professor with the Universidad de Las Americas, Ecuador.