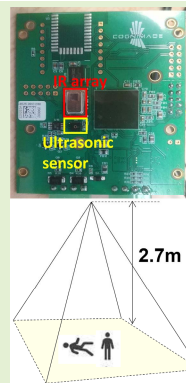


A Random Forest Approach to Body Motion Detection: Multisensory Fusion and Edge Processing

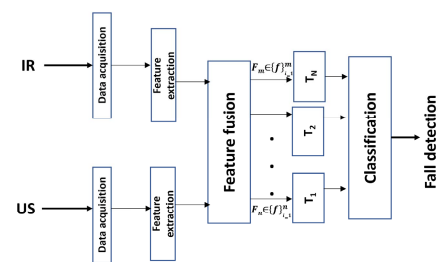
Sanaz Kianoush¹, Member, IEEE, Stefano Savazzi¹, Member, IEEE, Vittorio Rampa¹, Member, IEEE, Leonardo Costa¹, and Denis Tolochenko¹

Abstract—Low-complexity and privacy-respecting human sensing is a challenging task in smart environments as it requires the orchestration of multiple sensors, low-impact machine learning (ML) methods, and resource-constrained Internet of Things (IoT) devices. Client/server-based architectures are typically employed to support sensor fusion. However, these architectures need data to be moved to/from the cloud or data centers, which is contrary to the fundamental requirement of the IoT applications to limit costs, complexity, memory footprint, processing, and communication resources. In this article, we propose the design and implementation of an integrated edge device targeting human sensing for indoor smart spaces applications envisioned in Industry 5.0 applications. The proposed device implements the cumulative sum (CUSUM) method for data distillation from multiple sensors and adopts a low-complexity random forest algorithm (RFA) to sense and classify body movements: in particular, the device integrates both infrared (IR) and ultrasonic (US) sensors. This article discusses the benefits of the combined use of CUSUM and RFA methods against classical ML approaches in terms of accuracy, complexity, computing time, and storage. The proposed architecture and processing steps are validated experimentally by targeting the fall detection problem in a smart space environment. RFA reduces the complexity by at least three times compared to classical ML tools based on the analysis of space and time features (convolutional neural networks and long short-term memory): processing time is in the order of 0.1 s while accuracy is about 94%.

Index Terms—Edge processing, fall detection, Internet of Things (IoT), machine learning (ML), random forest, sensor fusion.



Edge based multisensory integrated device



I. INTRODUCTION

THE increasing digitalization in smart buildings and industrial environments, namely, Industry 4.0 (I4.0), is paired with the massive diffusion and deployment of heterogeneous Internet of Things (IoT) sensors, mostly exploited for

Manuscript received 25 October 2022; revised 16 December 2022 and 21 December 2022; accepted 21 December 2022. Date of publication 4 January 2023; date of current version 13 February 2023. This work was supported by the Regione Lombardia Tech-Fast POR-FESR Program. An earlier version of this paper was presented at the IEEE Sensors Virtual Conference, October 2021 [DOI: 10.1109/SENSORS47087.2021.9639706]. The associate editor coordinating the review of this article and approving it for publication was Prof. Jean-Michel Redoute. (Corresponding author: Sanaz Kianoush.)

Sanaz Kianoush, Stefano Savazzi, and Vittorio Rampa are with the Institute of Electronics, Information Engineering and Telecommunications (IEIT), Consiglio Nazionale delle Ricerche, 20133 Milan, Italy (e-mail: sanaz.kianoush@ieit.cnr.it; stefano.savazzi@ieit.cnr.it; vittorio.rampa@ieit.cnr.it).

Leonardo Costa and Denis Tolochenko are with Cognimade S.r.l., 20066 Melzo, Italy (e-mail: leonardo.costa@cognimade.com; denis.tolochenko@cognimade.com).

Digital Object Identifier 10.1109/JSEN.2022.3232085

automated applications [1]. In particular, the transition from Industry 4.0 to Industry 5.0 (I5.0) is currently supported by artificial intelligence (AI) systems to make human-centered smart environments more integrated, dynamic, adaptable, and connected. However, this makes also the environments inherently more complex, with large volumes of data being generated and processed.

Successful transition into AI-based I5.0 applications faces many challenges such as privacy (i.e., data ownership), security (i.e., data security and integrity), and complexity, as AI tools must be embedded into the IoT devices with limited computing, memory, and communication resources [2]. In addition, real-time human sensing through the orchestration of many heterogeneous sensors, which need to send continuously raw data to cloud servers, poses additional constraints, such as data privacy and integrity, as well as an efficient use of the generated data [3].

Fusion of data generated by multiple sensors, through processing and learning at the edge or intermediate devices, rather than on the cloud, is a fundamental, and practical,

solution [4]. However, due to the limited computational and communication capacity at the edge devices, it is also essential to apply low-complexity but high-accuracy designs [5] of the processing, learning, and inference stages. For example, data distillation methods, such as feature extraction and processing [4], must be implemented inside the IoT devices themselves to avoid moving large-size data over bandwidth- or energy-limited networks.

Through feature-based distillation, field devices, or sensors, producing data are responsible per part of data preprocessing as they share and distribute features rather than raw data. This must be accomplished not only to respect data ownership, integrity, and privacy but also to limit the data complexity by reducing the data dimensionality to a small set of features/parameters exchanged.

This article targets the design, implementation, and validation of an integrated edge device with real data for human sensing in indoor smart spaces. The device is equipped with multiple sensors and implements real-time sensor fusion, feature extraction, and edge processing methods optimized to achieve good accuracy with a significant reduction in memory, processing time, and complexity. In particular, this article proposes the adoption of a cumulative sum (CUSUM) approach for efficient data distillation from multiple sensors. It also compares different machine learning (ML) approaches to support joint body motion and fall detection applications. A decision tree classifier, based on the random forest algorithm (RFA), is proposed and compared with classical data-intensive neural network-based approaches in terms of algorithm complexity, memory and processing time requirements, and accuracy.

Fig. 1(a) and (b) shows the proposed sensor fusion architecture: the edge device is equipped with infrared (IR) and ultrasonic (US) sensors, preprocessing units (PUs), and a feature fusion center (FC). The IR and US sensors acquire data independently, namely, thermal and range information, and send them to the PUs for feature extraction. Each PU applies raw data distillation by implementing the CUSUM algorithm [30], [31], from which features are extracted and forwarded to the FC [implemented on the microcontroller unit (MCU)]. The FC performs feature selection and aggregation. It provides the results for the learning and decision steps based here on RFA. Feature evaluation and decision tree-based processing are performed to reduce the data complexity and mitigate the computational cost at the edge.

A. Related works

Human-centric smart building applications are part of Industry 5.0 trends and target the real-time human body monitoring in indoor/outdoor areas [7], [8]. Passive sensing of people/workers and monitoring of their health conditions can be based on wireless local area network (WLAN) signal processing tools, as reviewed in [9]. For example, a device-free fall detection algorithm, based on the analysis of ambient radio signals, is proposed in [10] by exploiting low-power IEEE 802.15.4 compliant devices already deployed in the monitored area for machine-to-machine (M2M) communication

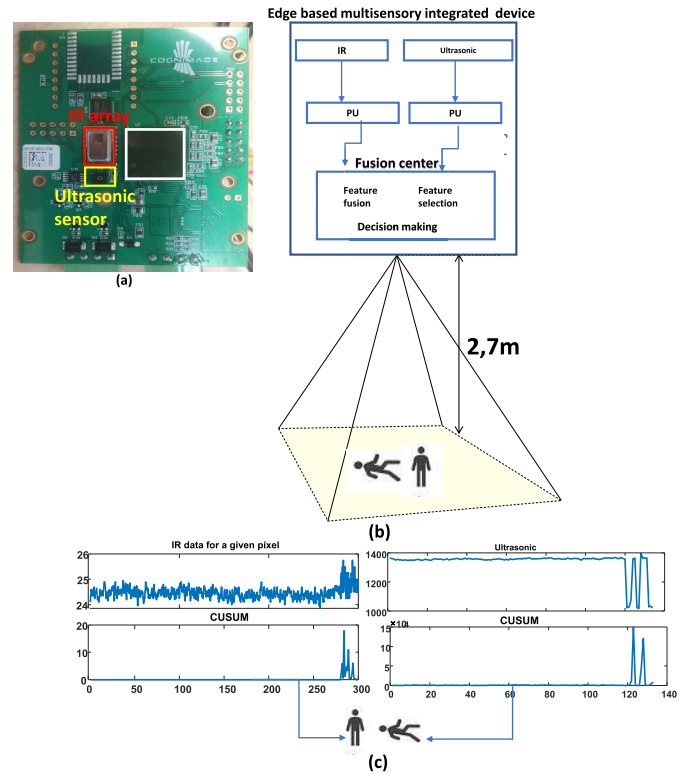


Fig. 1. Sensor fusion architecture. (a) Integrated sensor/edge device with both IR and US sensors, MCU, and DSP. (b) IoT edge device setup and layout for body and fall detection applications. (c) Examples of IR and US raw signals and the corresponding extracted CUSUM features.

purposes. Beside WLAN sensing, multisensor fall detection algorithms are analyzed by implementing information fusion of features extracted from experimental data collected by different sensors [11], namely, triaxial accelerometers, micro-Doppler radars, and depth cameras. However, with respect to this approach, the former does not require to wear any device; thus, it does not disclose any personal data (i.e., it is privacy-preserving by design).

In order to improve the accuracy and the efficient use of multiple sensors, a cloud-based multisensor fusion is proposed in [12], while in [13], a fusion algorithm is developed to combine range information from different radar sensors. In this case, a radar data fusion system is implemented for human gait estimation, in addition to fall detection. An approach capable of selecting the best fusion architecture (from predefined options) for a specified sensor dataset is also proposed in [14] where the sequential forward floating selection (SFFS) algorithm is exploited to reduce the features number and limit it to a given value.

Smart buildings and industrial environments resort to a massive deployment of resource-constrained IoT devices, sensors, and actuators. Sensors generate a large volume of data in real time, which is an appealing target for AI and data fusion systems [15]. However, deploying data-intensive ML models on such end devices is nearly impossible due to limited computing power, missing specialized AI processing units, and memory constraints. In some applications, the cost of building the computing/communication infrastructure and the

related fees to maintain it over time is also much higher than the benefits obtained from gathering and analyzing the same endpoint data. Beside costs, there are also concerns over bandwidth limitations and processing time, as data need to be sent to the cloud, while the analysis results must be returned to the endpoints in a timely manner.

Edge and fog computing methods [16], [17], [18] have been proposed to migrate computations from the data center near to the sensors producing the data or other colocated devices, thereby reducing the system latency, bandwidth occupation [19], and computing resources [20]. Processing on the edge allows also to perform inference near the source of the data; on the other hand, limited resource constraints prevent the adoption of data-intensive and/or complex models [21].

Existing works on ML-based activity recognition [23], [24] exploit long short-term memory (LSTM) [25] and convolutional neural networks (CNNs) [26] architectures as they take advantage of temporal and spatial features, respectively. Focusing on low-complexity designs, RFA-based methods are expected to provide comparable accuracy to LSTM and CNN while being more easily implemented on embedded devices with limited computational resources [27].

Running ML tools on embedded edge devices brings new challenges that include: 1) data distillation, to extract low-dimensional features from raw training data; 2) ML model optimization for resource-constrained computing environments; 3) real-time performance (latency and computing time); and 4) accuracy versus complexity tradeoff analysis.

To address these issues, prior works about ML on edge devices focused on deep learning (DL) methods [16], [17], [18]. Typical approaches propose to keep the ML model size small enough by using few trainable parameters, minimizing the number of computations [15], adopting quantization, sparsification, pruning [22], and DNN partitioning [28]. However, these methods are still debatable in dynamic network scenarios with constrained resource devices since they require additional preprocessing steps or increased communications with edge/cloud servers due to offloading or specific hardware support to maintain a reasonable accuracy.

Unlike previous studies, this article proposes a data-agnostic architecture that combines a CUSUM technique [29], [30] for data distillation, followed by RFA processing. CUSUM implements a semiblind change point detection and is adopted for feature extraction and fusion of raw sensor data. CUSUM was originally proposed for anomaly detection [30] and recently adopted also for data segmentation [6], [31]. Compared to raw data, CUSUM produces low-dimensional features [6] suitable for resource-constrained devices. These features are fed to an RFA model designed to classify human worker motions.

B. Contributions

This article proposes the design, implementation, and validation of an edge device that integrates IR and US sensors targeting body motion and activity recognition for smart spaces applications. The device supports preprocessing and feature-based data fusion techniques optimized for implementation on resource-constrained hardware. Rather than sharing, and directly processing, the raw data from the IR and US

sensors, the proposed technique performs low-dimensional feature extraction by exploiting, for the first time, a change point detector tool that is based on the CUSUM approach. Features are then used as inputs for body motion detection that is performed by an ad hoc RFA. The approach is shown to be more efficient compared to classical DL-based approaches in terms of complexity, computing time, storage, and accuracy. Besides, the proposed approach does not need data to be moved on the cloud and thus protects data ownership, integrity, and safety.

The proposed approach is analyzed in terms of complexity, namely, the expected number of operations to produce a detection output, computing time for training/inference, memory footprint, and accuracy. The algorithm also performs the selection of the best pool of features to maximize the detection performance targeting the human body fall detection problem.

This article extends [31] with novel contributions summarized as follows.

- 1) An edge-based feature processing model that integrates multiple sensors, data distillation, and data fusion is proposed for body motion and activity recognition applications. The CUSUM method is used to extract features (i.e., for data distillation), while feature selection, grouping, and fusion are based on the RFA method optimized to recognize critical body motion events.
- 2) An edge-based integrated board is designed to implement the proposed algorithm: it includes the sensors, the PUs, and the FC. The multisensor board is equipped with a low spatial resolution IR sensor composed by 64 thermopile elements, arranged as a squared array of 8×8 IR pixels, and a US sensor for US ranging. Hardware and sensor integration are also discussed.
- 3) The approach is validated experimentally in an indoor environment. In particular, the study focuses on body walk vs fall classification, as critical in smart spaces applications. The RFA method is also compared against popular data-intensive ML designs based on CNN and LSTM [32], [33].

This article is organized as follows. Section II discusses the CUSUM-based sensor fusion model, the feature extraction, and the feature reduction method, tailored for IR arrays and US sensors. Section III describes the proposed RFA method for activity recognition (i.e., body walking versus falling). Section IV analyzes a case study for fall detection in an indoor smart space environment. Accuracy-complexity tradeoffs, training, and processing time are analyzed in detail. Finally, Section V draws the conclusions and discusses current limitations and future works.

II. EDGE-BASED SENSOR FUSION MODEL: SENSOR DATA AND FEATURE PROCESSING

This section analyzes the data fusion architecture targeting the IR and US sensors integration. It includes the data acquisition and the feature fusion models for the human sensing problem. Among various sensors and IoT technologies, thermal vision systems, based on low-cost IR array sensors, allow to track thermal signatures induced by moving people.

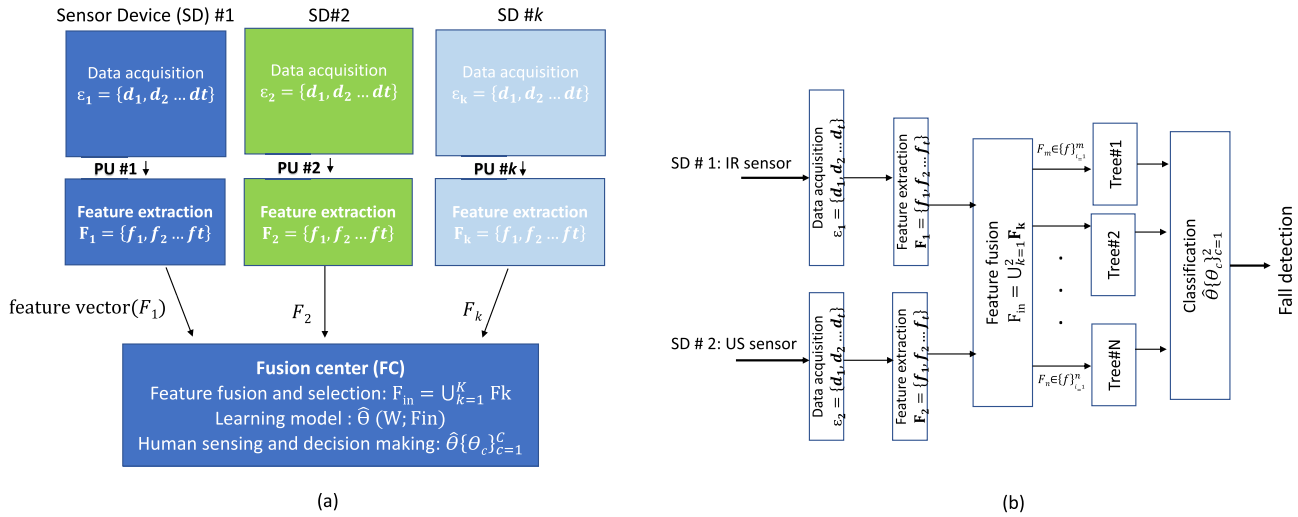


Fig. 2. Multifeature fusion model and architecture: (a) general multifeature fusion model for human sensing using K sensors and (b) specific multifeature fusion architecture for fall detection applications based on IR and US sensors.

Moreover, US sensors estimate the distance from obstacles or other people to improve safe indoor navigation. Unlike contact tracing applications based on short-range communications, IR and US-based sensing systems are passive, as they do not need neither the cooperation of the subject(s) nor the use of any wearable device. Thus, they do not pose any threat to user privacy and are also suitable for deployment in large spaces such as smart buildings or industrial workplaces [31].

Fig. 2(a) shows the general multifeature fusion model and the components of the integrated multisensor platform, namely, the edge device, and its functionalities. The integrated edge device includes the sensor devices (SD_k), the corresponding processing units (PU_k), and the FC. Each sensor device produces the raw data that are fed to the PUs (Section II-A), while the PUs implement a change detector algorithm and extract the local features obtained from the CUSUM metric (described in Section II-B). Finally, the FC learns a global RFA model for body motion detection using the local CUSUM features. In addition, it also performs the feature selection, grouping, and fusion stages.

Human body movements may be captured by exploiting both the temperature perturbations in the readings of the IR array sensor SD_1 and the changes in the distance measurements produced by the US sensor SD_2 . Fig. 2(b) shows the setup and the classification problem targeting body motion detection (in our case, the fall versus nonfall events) using data collected by both sensors and their corresponding features. During the learning phase of the RFA model, the FC selects an optimal subset of features (i.e., number and types) that provide the maximum accuracy and the lower training time. As described in Section III, each tree makes the local decision using the selected features subset in the training phase. A final decision is made by major voting, considering all individual trees.

In the following, we introduce the statistical model for the raw data captured by the IR array and the US sensor

sources [31], [34]. Notice that the proposed model and method can also serve as a general framework for multisensor deployments. Data acquisition and CUSUM-based feature extraction steps are described as well.

A. Multisensor Data Processing

IR sensors typically consist of a single or multiple arrangement (i.e., array) of thermopile elements equipped with lens, mostly deployed in indoor scenarios to estimate the temperature of the environment and the objects/people herein contained [34], [35]. For instance, passive IR (PIR) sensors are widely employed in consumer security systems [36], also in conjunction with radar sensors [37], to detect the people presence in the field of view (FoV) of the sensor. However, recently, IR arrays have been employed not only for human body occupancy detection but also for localization and tracking applications [34].

Typical US sensors are noncontact micro-electromechanical system (MEMS) devices that provide only range (i.e., distance) information about targets inside the sensor FoV. Since they do not extract any visual image of the target(s), as in the case of low-resolution IR sensor arrays, both sensors are privacy-preserving and can also be fused together to improve the detection accuracy.

1) *IR Sensor Array*: Low-cost and low-resolution IR sensor arrays, presently available on the market, usually consist of M thermopile elements arranged as a single element ($m = 1$) or multiple elements ($m = 1, \dots, M$) in 1-D linear or 2-D grid, whose format depends on the specific IR device. In our case, the IR array, namely, the grid eye [38], acquires raw thermal images organized as 2-D frames of 8×8 pixels ($M = 64$), while the frame sampling frequency of the array is set to 10 Hz. The IR array includes a built-in silicon lens having an FoV equal to 60° and features 8 pixels vertically and 8 pixels horizontally having an opening angle of about 7.5° per pixel (i.e., with very low spatial resolution). This corresponds to the approximated full-width at half-maximum

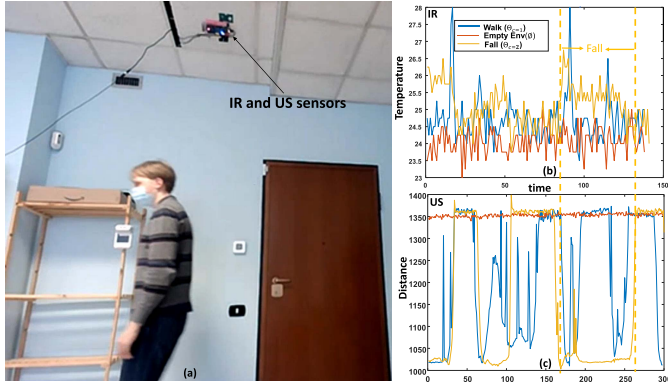


Fig. 3. Human body motion detection: (a) indoor deployment of the ceiling-mounted sensors board; (b) IR sensor readings $\varepsilon_{t,m}^{(IR)}$ for the IR array element with index $m = 30$ and acquisition time equal to 15 s; and (c) US sensor readings $\varepsilon_t^{(US)}$ during walk and fall activities with respect to the empty environment (Env) for the acquisition time equal to 30 s.

(FWHM) value. The sensor is designed for IR wavelengths in the range of 5–13 μm with a noise temperature equivalent difference (NETD) equal to 0.16 $^{\circ}\text{C}$ according to [38].

IR arrays can be wall- or ceiling-mounted [39]. However, without any loss of generality, we consider a ceiling-mounted device, as shown in Fig. 3(a). During the experiments, a person enters in an area of about 12 m^2 and walks or falls during a period of approximately 30 s for several times. Fig. 3(b) shows an example of human body motion captured by the m th thermopile of the IR array sensor (actually, the one with index $m = 30$) during fall and walk activities with respect to the empty environment taken as a reference. As confirmed in the example, when a person enters in the area and walks (blue color), the temperature readings with respect to the empty environment increase, while the target moves in the surroundings of the considered thermopile element FoV (see the corresponding peaks in the figure). On the contrary, the samples highlighted in yellow indicate a person that loses the balance and falls. These informative patterns are captured and used to identify walk and fall activities from the empty environment case.

The sensor readings at time t are collected in the vector $\mathbf{\varepsilon}_t^{(IR)} = [\varepsilon_{t,1}^{(IR)}, \dots, \varepsilon_{t,M}^{(IR)}]^T$ where each element $\varepsilon_{t,m}^{(IR)}$ corresponds to a noisy thermal measurement obtained from the m th thermopile component. For the example of Fig. 4, the detector array acquires raw thermal IR images organized as 2-D frames corresponding to the empty [Fig. 4(a)] or occupied space, where a person walks [Fig. 4(b)] or falls [Fig. 4(c)], respectively. The sensor readings $\mathbf{\varepsilon}_t^{(IR)}$ depend linearly on the body occupancy vector $\mathbf{r}_t = [r_{t,1}, \dots, r_{t,m}, \dots, r_{t,M}]^T$, whose elements $r_{t,m} \in \{0, 1\}$ provide binary information about the presence, or absence, of the subject at time t as detected by the m th thermopile element. In the following, we also define the human body state $\Theta_{t,m}$ at time t to represent the activity of the subject in the environment (i.e., fall versus nonfall state) detected by the m th thermopile element at time t . The vector $\mathbf{\varepsilon}_t$ is modeled as

$$\mathbf{\varepsilon}_t^{(IR)} = \mathbf{H} \cdot \mathbf{r}_t + \mathbf{w}_t = \sum_{m=1}^M \mathbf{h}_m(\Theta_{t,m}) \cdot r_{t,m} + \mathbf{w}_t \quad (1)$$

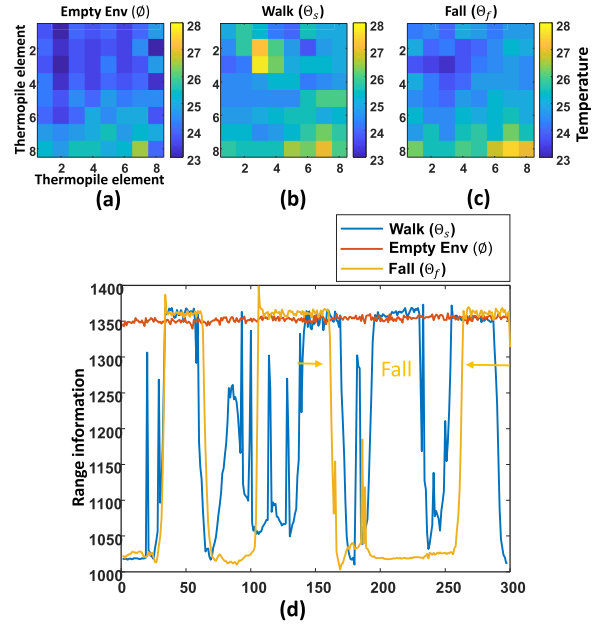


Fig. 4. Examples of IR and US readings in a typical indoor scenario. The 2-D images of the IR sensor readings correspond to: (a) empty space, (b) person walking, and (c) person falling. In (d), the range information from the US sensor readings corresponds to the time evolution for a person walking or falling in the monitored area with respect to the empty environment.

where, for each k th element of $\mathbf{\varepsilon}_t^{(IR)}$, it is $\varepsilon_{t,k}^{(IR)} = \sum_{m=1}^M h_{k,m}(\Theta_{t,m}) \cdot r_{t,m} + w_{t,k}$. The $M \times M$ matrix $\mathbf{H} = \mathbf{H}(\Theta_{t,m}) = [\mathbf{h}_1, \dots, \mathbf{h}_m, \dots, \mathbf{h}_M]$ collects all the M thermal signatures \mathbf{h}_m [39] that are represented by the column vector $\mathbf{h}_m = [h_{1,m}, \dots, h_{M,m}]^T$ of size $M \times 1$. These signatures describe the pattern of temperature increases induced by a body state $\Theta_{t,m}$. Clearly, in the absence of the target, i.e., for $\Theta_{t,m} = \emptyset$, it is $\mathbf{h}_m(\Theta_{t,m}) = \mathbf{0}$, where $\mathbf{0}$ is the zero vector of size $M \times 1$. The background temperature \mathbf{w}_t is described by the column vector $\mathbf{w}_t = [w_{t,1}, \dots, w_{t,M}]^T$ that conveys information about noisy heat sources that are not caused by body movements but still characterize the empty space. As clarified in the following, the background temperature in the absence of the target is modeled by an autoregressive (AR) model [6].

2) *US Sensor*: The adopted US sensor is a time-of-flight (ToF) sensor, namely, the CH201 from TDK [40], which internally includes a MEMS-based piezoelectric micromachined ultrasonic transducer (PMUT) [41] and a system-on-chip (SoC) unit for range preprocessing. The US device uses the PMUT to transmit (and receive) short pulses of sound waves through the air [31] (and references herein), with an FoV of about 45° . The US sensor range readings are collected in the vector $\mathbf{\varepsilon}_t^{(US)} = [\varepsilon_{t-K+1}^{(US)}, \dots, \varepsilon_t^{(US)}]^T$ that includes K consecutive samples $\varepsilon_{t-k}^{(US)}$ up to time t . As shown in [31], the round-trip time $t_s(\Theta_{t,k})$, due to the sound waves' propagation from and back to the PMUT, depends on the human body state $\Theta_{t,k}$ and it is measured by the built-in SoC as

$$\varepsilon_{t,k}^{(US)} = c_{ac} \cdot t_s(\Theta_{t,k}) / 2 = \|M - S(\Theta_{t,k})\|_2 \quad (2)$$

where $c_{ac} = 343$ m/s is the speed of sound at the room temperature (i.e., 20 °C), $M = [x, y]^T$ is the known transducer's position, and $S(\Theta_{t,k}) = [x_k, y_k]^T$ is the position of the human body in state $\Theta_{t,k}$.

Notice that the detection performance typically suffers from disturbance degradation in the presence of adverse environmental factors such as non-line-of-sight (NLOS) and multipath propagation effects [42] that typically induce biased received range data. However, these phenomena are generally unavoidable in real deployments, due to the inherent nature of the sound waves used for ranging that are prone to be reflected, refracted, and diffracted when encountering different kinds of obstacles (e.g., walls, ceiling, and ground) and even human beings that are present in the monitored area [43], [44].

B. Change Detection and CUSUM Features

In what follows, we investigate the distillation and segmentation algorithms applied to the IR and US sensors' raw data. To this aim, we exploit the CUSUM method (see Fig. 2) to extract the features related to the changes in the sensor readings due to specific human body motion activities.

Considering here an indoor environment without targets ($\Theta_{t,m} = \emptyset$), the sensor readings time series $\boldsymbol{\varepsilon}_t^{(\text{IR})}$ and $\boldsymbol{\varepsilon}_t^{(\text{US})}$, obtained from the IR and US sensors, are here modeled by a p th-order AR model [6]. More specifically, the signal produced at time t by the m th thermopile element of the IR array is modeled iteratively as

$$\varepsilon_{t,m}^{(\text{IR})}(\boldsymbol{\theta}_{0,m}^{(\text{IR})}) = \mathbf{b}_m^T \cdot \boldsymbol{\varepsilon}_{t-1:t-p,m}^{(\text{IR})} + \mathbf{n}_{t,m}^{(\text{IR})} \quad (3)$$

where the column vector $\boldsymbol{\theta}_{0,m}^{(\text{IR})} = [\mathbf{b}_m, \sigma_m]^T$ of size $p+1$ defines the AR parameters and the column vector $\boldsymbol{\varepsilon}_{t-1:t-p,m}^{(\text{IR})} = [\varepsilon_{t-1,m}^{(\text{IR})}, \dots, \varepsilon_{t-p,m}^{(\text{IR})}]^T$ of size p collects the previous p IR samples of the m th element of the IR array, while the Gaussian-distributed white noise $\mathbf{n}_{t,m}^{(\text{IR})} \sim \mathcal{N}[0, \sigma_m^2]$ models the innovation term $\mathbf{n}_{t,m}^{(\text{IR})}$ with zero mean and variance σ_m^2 . Likewise, focusing on the US sensor, the AR model is defined as

$$\varepsilon_t^{(\text{US})}(\boldsymbol{\theta}_0^{(\text{US})}) = \mathbf{b}_u^T \cdot \boldsymbol{\varepsilon}_{t-1:t-p}^{(\text{US})} + \mathbf{n}_t^{(\text{US})} \quad (4)$$

where the column vector $\boldsymbol{\theta}_0^{(\text{US})} = [\mathbf{b}_u, \sigma_u]^T$ of size $p+1$ includes the background-specific AR parameters and the column vector $\boldsymbol{\varepsilon}_{t-1:t-p}^{(\text{US})} = [\varepsilon_{t-1}^{(\text{US})}, \dots, \varepsilon_{t-p}^{(\text{US})}]^T$ of size p collects the previous p US samples, while $\mathbf{n}_t^{(\text{US})} \sim \mathcal{N}[0, \sigma_u^2]$ models the residual range as a zero-mean Gaussian white stochastic process with standard deviation σ_u . In this article, we assume $p = 1$, while the proposed AR models are validated experimentally in Section IV.

A change in the environment, e.g., due to an object or a person, standing or moving in the area covered by the IR and the US sensors, alters the corresponding AR parameters, represented by the matrix $\boldsymbol{\Psi}_0 = [\boldsymbol{\theta}_{0,1}^{(\text{IR})}, \dots, \boldsymbol{\theta}_{0,M}^{(\text{IR})}, \boldsymbol{\theta}_0^{(\text{US})}]$ of size $(p+1) \times (M+1)$. The parameters $\boldsymbol{\Psi}_0$ are replaced with the new, but unknown [31], ones $\boldsymbol{\Psi}_1 = [\boldsymbol{\theta}_{1,1}^{(\text{IR})}, \dots, \boldsymbol{\theta}_{1,M}^{(\text{IR})}, \boldsymbol{\theta}_1^{(\text{US})}]$. Detection of changes in the AR parameters is implemented iteratively by monitoring the CUSUM function [30]. For a

given inspection interval $[t-T, t]$ of duration T , the CUSUM g_t is defined at time t for both IR array and US sensors as

$$g_{t,m}^{(\text{IR})} = \frac{1}{T} \sum_{j=t-T}^t (\ell_{j,m})^+ \quad (5)$$

and

$$g_t^{(\text{U})} = \frac{1}{T} \sum_{j=t-T}^t (\ell_j)^+ \quad (6)$$

respectively, where the operator $(\cdot)^+$ is defined as $(\cdot)^+ = \max[0, \cdot]$, while the log-likelihood ratio terms $\ell_{j,m}$ and ℓ_j are defined as

$$\begin{aligned} \ell_{j,m}(\boldsymbol{\theta}_{0,m}^{(\text{IR})}) &= \sup_{\boldsymbol{\theta}_{1,m}^{(\text{IR})} \in \Psi_1} \left(\ln \left[\frac{\Pr_{\boldsymbol{\theta}_{1,m}^{(\text{IR})}}(\boldsymbol{\varepsilon}_{t,m}^{(\text{IR})} | \boldsymbol{\varepsilon}_{t-1|t-q-1,m}^{(\text{IR})})}{\Pr_{\boldsymbol{\theta}_{0,m}^{(\text{IR})}}(\boldsymbol{\varepsilon}_{t,m}^{(\text{IR})} | \boldsymbol{\varepsilon}_{t-1|t-q-1,m}^{(\text{IR})})} \right] \right) \\ \ell_j(\boldsymbol{\theta}_0^{(\text{US})}) &= \sup_{\boldsymbol{\theta}_1^{(\text{US})} \in \Psi_1} \left(\ln \left[\frac{\Pr_{\boldsymbol{\theta}_1^{(\text{US})}}(\boldsymbol{\varepsilon}_t^{(\text{US})} | \boldsymbol{\varepsilon}_{t-1|t-q-1}^{(\text{US})})}{\Pr_{\boldsymbol{\theta}_0^{(\text{US})}}(\boldsymbol{\varepsilon}_t^{(\text{US})} | \boldsymbol{\varepsilon}_{t-1|t-q-1}^{(\text{US})})} \right] \right) \end{aligned} \quad (7)$$

respectively. The supremum (sup) is computed over the (unknown) parameter set $\boldsymbol{\theta}_{1,m}^{(\text{IR})}$ and $\boldsymbol{\theta}_1^{(\text{US})} \in \Psi_1$, while $\ln[\cdot]$ functions are the log-likelihood ratios for $\boldsymbol{\theta}_{1,m}^{(\text{IR})}$ and $\boldsymbol{\theta}_{0,m}^{(\text{IR})}$ and $\boldsymbol{\theta}_1^{(\text{US})}$ and $\boldsymbol{\theta}_0^{(\text{US})}$ pairs, respectively. For small changes of the ε_t dynamics with respect to the empty environment, as typically observed during body movements, the terms $\ell_{j,m}, \ell_j$ in (7) simplify to [30]

$$\begin{aligned} \ell_{j,m} &\approx v_m \mathbf{u}_{\text{IR}}^T \cdot \mathbf{z}_{j,m}(\boldsymbol{\theta}_{0,m}^{(\text{IR})}) \\ \ell_j &\approx v \mathbf{u}_{\text{US}}^T \cdot \mathbf{z}_j(\boldsymbol{\theta}_0^{(\text{US})}) \end{aligned} \quad (8)$$

where the constant terms $v, v_m > 0$, and $\mathbf{z}_j, \mathbf{z}_{j,m}$ are the score function vectors of size $(p+2) \times 1$ defined as in [30]. The unitary vectors \mathbf{u}_{IR} and \mathbf{u}_{US} indicate the predominant directions of the change that maximize the impact of the perturbations on the selected model parameters (i.e., AR parameters and residual standard deviation). In our case, we assume that $v = v_m = 0.2$ and an AR-1 (i.e., $p = 1$) model; in addition, $\mathbf{u}_{\text{IR}} = \mathbf{u}_{\text{US}} = [\sin(\alpha), \cos(\alpha)]^T$, with $\alpha = (\pi/20)$. The score functions further simplify as

$$\begin{aligned} \mathbf{z}_{j,m} &= \left[\frac{1}{\sigma_m^2} \mathbf{n}_{t,m}^{(\text{IR})} \boldsymbol{\varepsilon}_{t-1,m}^{(\text{IR})}, \frac{1}{\sigma_m} \left(\left[\frac{\mathbf{n}_{t,m}^{(\text{IR})}}{\sigma_m} \right]^2 - 1 \right) \right]^T \\ \mathbf{z}_j &= \left[\frac{1}{\sigma_m^2} \mathbf{n}_t^{(\text{US})} \boldsymbol{\varepsilon}_{t-1}^{(\text{US})}, \frac{1}{\sigma_m} \left(\left[\frac{\mathbf{n}_t^{(\text{US})}}{\sigma_u} \right]^2 - 1 \right) \right]^T. \end{aligned} \quad (9)$$

The CUSUM results from the IR and US sensors $g_{t,m}^{(\text{IR})}, m = 1, \dots, M$, and $g_t^{(\text{US})}$ are then transferred to the FC, and the human body state Θ_t is recognized using both fused feature terms $g_{t,m}^{(\text{IR})}$ and $g_t^{(\text{US})}$ with

$$F(\hat{\Theta}_t) = \left\{ g_{t,m}^{(\text{IR})}; g_t^{(\text{US})} \right\}. \quad (10)$$

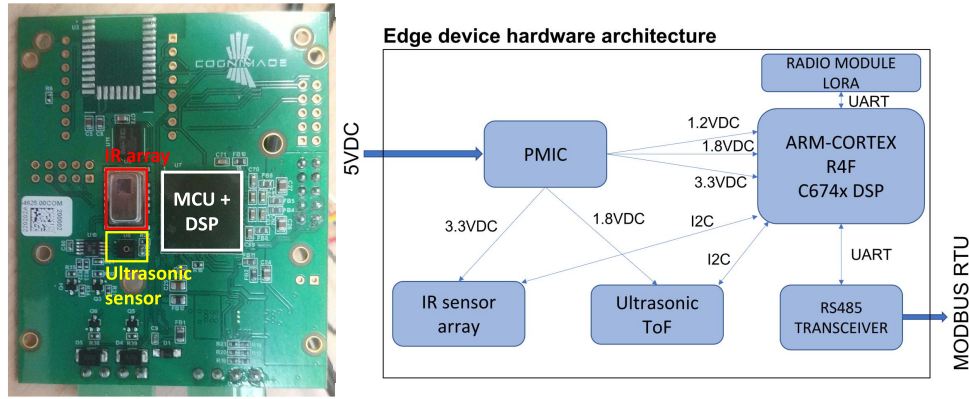


Fig. 5. Edge device (EnviRadar prototype) and hardware architecture. Buses and power supply lines are detailed.

Finally, the proposed RFA method uses the features (10) as input for classification of body motion (fall detection).

Note that the CUSUM-based processing is a proper solution for handling heterogeneous sensor inputs (i.e., it is not specific to IR and US signals) as it can accurately capture environmental changes by analyzing multidimensional time series regardless of the raw data sources [6].

C. Edge-Based Board: Hardware and Sensors integration

In this section, we present the edge-based integrated board that has been used to experimentally verify the proposed feature fusion methods. The edge device is assumed to be ceiling-mounted, as shown in Fig. 3(a). The developed edge board is shown in Fig. 5 that highlights also the specific hardware architecture and the related design details. The board integrates the following main components.

- 1) A power management integrated circuit (PMIC) that provides four voltage levels as outputs (i.e., 1.0, 1.2, 1.8, and 3.3 V) required by the different sensors, the main microcontroller, and the other ancillary devices.
- 2) The ARM-Cortex R4F microcontroller running at 200 MHz and the digital signal processor (DSP) TMS320C674x at 600 MHz with shared RAM for computations, namely, features extraction and fusion.
- 3) The US sensor CH201 [40], [41] interfaced to the ARM Cortex R4F via the I2C bus that supports distance measurement up to 5 m, with a customizable FoV up to 180 °C but limited here to 45 °C.
- 4) The IR array grid eye [38] with 0.25 °C temperature resolution and 10-Hz sampling rate. Likewise, the US sensor is interfaced with the microcontroller ARM Cortex R4F by exploiting the I2C bus.

Beside the above components, an RS485-compliant 20-Mb/s interface is used to send the extracted features $g_{t,m}^{(IR)}$ and $g_t^{(US)}$ to external units (e.g., ModBus devices) for further processing. The transceiver is interfaced to the ARM Cortex R4F by exploiting its embedded UART. The device may also be equipped with an optional module supporting LoraWAN radio for remote data sharing. Even in this case, the LoraWAN module is interfaced to the ARM Cortex R4F through another

embedded UART. In Section IV, we will exploit the RS485 interface for data extraction, while the LoraWAN radio module will not be considered.

III. RANDOM FOREST MODEL AND DECISION-MAKING FOR HUMAN SENSING

This section discusses the human body fall detection problem by exploiting the proposed multisensory fusion setup. Human body motions are detected through the CUSUM features that discriminate a body fall event from a (safe) walking activity of a person inside the monitored space. In what follows, we describe the classification problem and the adopted RFA method based on the fused multisensory features.

A. Random Forest and Decision Tree Ensemble Model

Decision trees are frequently encountered in ML applications and sometimes used also for classification and regression [45] on resource-constrained edge devices [33], [46]. A decision tree is represented in general as a function $\mathcal{T} : \mathbb{R}^n \rightarrow \mathbb{R}$ on an n -dimensional feature space. It is defined by recursively partitioning the input feature space $F \in \mathbb{R}^n$ into regions and implementing a binary test in each resulting region. A function can be represented by a tree that consists of nodes (representing a test) and branches that show the specific test outcomes. Classification is typically based on the analysis of the paths in the tree. One way to improve the performance of decision tree classifiers is to combine the responses from multiple trees. In [47], a set of decision trees are trained using a random subset of the input features. This has the effect of decorrelating the outputs of the individual trees in the forest, combining the advantages of bagging and random selection [48], and thus improving the performance. RFA is comparatively reported to perform faster than boosting and bagging and it is robust to outliers and noise. It is also an effective solution to handle large datasets and missing data as often encountered in sensor fusion applications [49].

The RFA model \mathcal{R} is described as an ensemble of N decision trees $\mathcal{R} = \{\mathcal{T}_i\}_{i=1}^N$ where each classifier tree \mathcal{T}_i contributes with a single vote for the assignation of the most frequent class to the input vector F . For the proposed sensor

fusion setup detailed in Section II, we collect $M + 1 = 65$ CUSUM terms where 64 are obtained from the IR sensor array (8×8) [50] and 1 from the US sensor readings. Next, we select a group of $k_f \leq M + 1$ features targeting the fall detection accuracy of 0.9. The feature group serves as input feature space F in (10) for \mathcal{R} . Finally, the RFA [51] uses f randomly selected subset of features $F_i \in F$ as an input to each tree, while each tree grows independently using the C4.5 algorithm [52].

B. Motion Classification: Body Fall Versus Walking Detection

In the following, we tackle the problem of body fall versus walking detection. The goal is to recognize the human body state $\hat{\Theta}_t \in \{\Theta_s, \Theta_f\}$ at time t , where Θ_s represents a subject in a safe state, i.e., walking/moving inside the coverage area, and Θ_f indicates a subject falling, comprising of prefall and fall event.

The change detector scheme described in Section II-B is used first to detect the body presence: the CUSUM features $F(\hat{\Theta}_t)$ in (10) are thus computed continuously with an inspection interval of $T = 5$ samples for both sensors (corresponding to 1 s for $T_s = 200$ ms sampling time). Motion events are segmented using two different thresholds applied to the IR array sensors $g_{t,m}^{(IR)}$ and the US $g_t^{(U)}$ CUSUM functions and optimized as shown in [31]. Segmented CUSUM features, indicating the presence of a subject, are then used as an input to the RF algorithm for classification.

The human body state at time t , $\hat{\Theta}_t$, is obtained by majority voting over local decisions $\{\hat{\Theta}_{i,t}\}_{i=1}^N$, where $\hat{\Theta}_{i,t} = \mathcal{T}_i(F_i)$ is the class prediction of the i th tree in the forest using the input feature subset F_i obtained at time t . Fall state Θ_f is an unpredictable event leading the subjects to lose the balance and rest on the ground/floor. As described above, a fall is detected through major voting over the N class predictions $\hat{\Theta}_{i,t}$, obtained from each RFA tree, namely, $\hat{\Theta}_t = \Theta_f$ iff

$$\sum_{i=1}^N \mathbf{1}_{\Theta_f}(\hat{\Theta}_{i,t}) > \sum_{i=1}^N \mathbf{1}_{\Theta_s}(\hat{\Theta}_{i,t}) \quad (11)$$

where $\mathbf{1}_y(x)$ is the indicator function, namely, $\mathbf{1}_y(x) = 1$ if $x = y$ and $\mathbf{1}_y(x) = 0$, otherwise.

IV. CASE STUDY IN A SMART BUILDING ENVIRONMENT

The chosen experimental setup described above is placed in an indoor smart space and employed for safety operations. Data acquisition, feature extraction, fusion, and, finally, classification steps are discussed with special focus on fall detection applications. The multisensor device shown in Fig. 5 is ceiling-mounted (2.7 m height) and equipped with one US sensor and one IR array module.

During the on-field tests, first, we collect data corresponding to the empty environment, $\Theta_t = \emptyset$, and then, we consider two scenarios: 1) a person entering inside the area, walking for 30 s and exiting from the room, and 2) a person entering the area, falling down on the floor and then staying on the ground for 30 s. Considering this last case, the sensors are able to track both the acceleration toward the floor, the shock between the

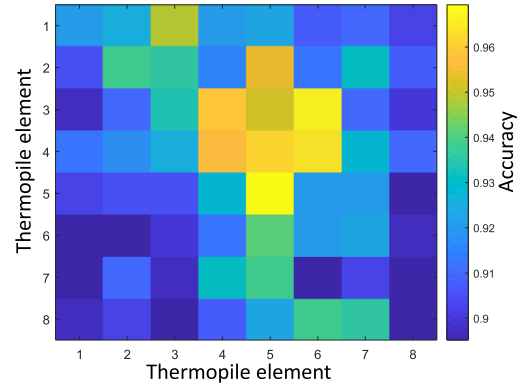


Fig. 6. Accuracy map of fall detection based on data collected from IR sensor for each individual thermopile element using RFA. The number of trees is set to $N = 25$.

faller and the floor (pre-fall activities) and the body lying on the ground (fall) [10].

A. Fall Detection Performance

First, we analyze the performance of the RFA-based human state classification using, alternatively, only one of the two sensors (i.e., IR or US). Next, we analyze the performance of the fusion algorithm. This is useful to compare the accuracy results of each sensor before and after the data fusion with features obtained from CUSUM. We define k_f and f as the number of input features and the size of the feature subsets used by each tree \mathcal{T}_i , respectively. Finally, we consider the impact of feature fusion and perform the complexity analysis.

1) *IR Sensor Array*: Targeting the IR sensor array, Fig. 6 shows the fall detection accuracy observed for each thermopile element (here represented as pixels) using the proposed RFA model $\mathcal{R} = \{\mathcal{T}_i\}_{i=1}^N$ with $N = 25$ trees. As shown in this example and usually found in almost all measurements, the pixels located in the center of the 2-D array show the larger accuracy. The above observation suggests that the selection of the best zone of pixels before implementing the RFA, see the features in (10), can provide marginal performance degradation in exchange for a significant reduction of the computational time by using less input features. As shown in Fig. 7, we define the zones $z = 1, \dots, 5$ as groups of pixels corresponding to colocated thermopile elements. Thus, the goal is to select the zone providing real-time activity recognition results with the largest accuracy. Fig. 7(a) shows the zones and the corresponding fall detection accuracy for each individual thermopile element. Focusing on the zones $z = 3$ and $z = 5$, Fig. 7(b) and (c) presents the accuracy results using a varying number of input features (k_f) obtained from the corresponding zones and a different number of trees N . In this case, each tree chooses the subsets of features randomly with size $f \leq (M)^{1/2} = 8$. The results confirm that pixel selection improves the fall detection accuracy, while the input features obtained from the thermopile elements located in the central zone $z = 5$ give the best performance.

Fig. 8 now summarizes the fall detection accuracy in terms of the feature subsets size f and the number N of RFA trees.

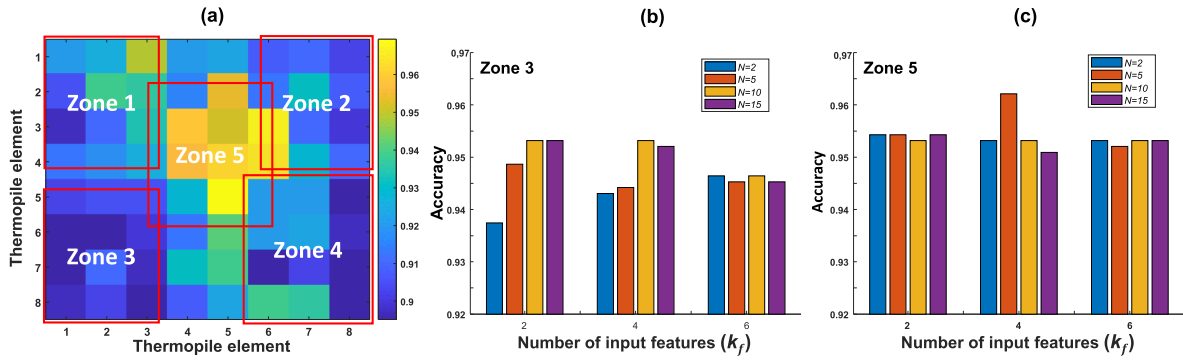


Fig. 7. Accuracy results of fall detection based on the selected input features: (a) thermopile accuracy map with five zones ($z = 1, \dots, 5$), (b) accuracy results for the zone $z = 3$, and (c) for the zone $z = 5$ in terms of different number N of trees T_i .

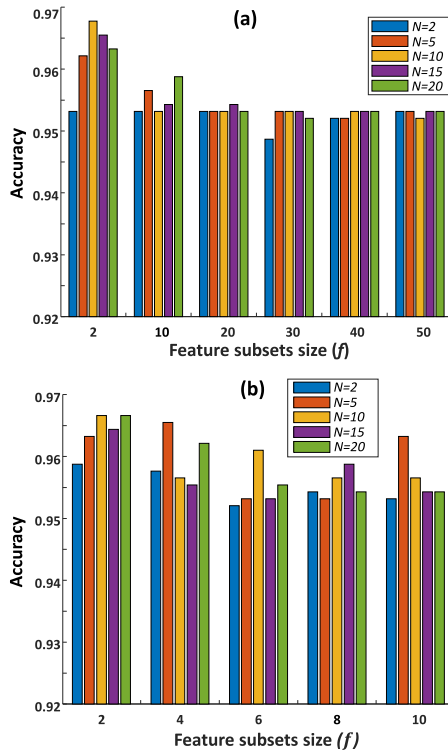


Fig. 8. Tradeoff among accuracy of fall detection results based on IR sensor data, feature subsets size f , and number N of trees T_i (colored bars): (a) feature subsets size ranging between $f = 2$ and $f = 50$ and (b) feature subsets size between $f = 2$ and $f = 10$.

The number of input features k_f is now fixed to $k_f = M = 64$. More specifically, Fig. 8(a) highlights the RFA accuracy for large values of f in the set $\{2, 10, 20, 30, 40, 50\}$, while Fig. 8(b) focuses on the accuracy results for small values of f , in the set $\{2, 4, 6, 8, 10\}$. Setting f large enough increases the probability that the N tree ensemble learns a representative subset of features and improves generalization. On the other hand, as clarified in the following, increasing the number of features too much would penalize the computational cost in exchange for marginal accuracy improvements (e.g., for $f > 20$ in the figure). For the same scenario, we also observe that increasing the number of trees does not always improve the fall detection accuracy: in particular, accuracy improvements

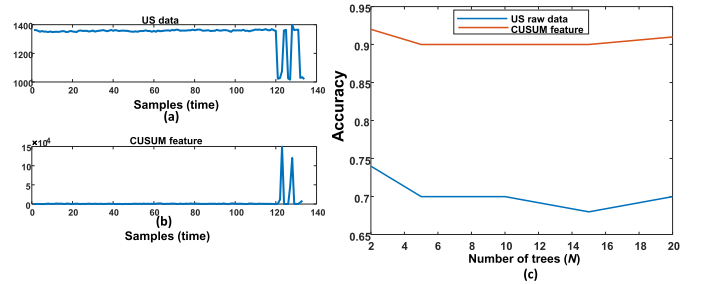


Fig. 9. Examples of raw US data, US CUSUM features, and accuracy tradeoff versus the number of trees for raw data and CUSUM features: (a) US raw data, (b) extracted US CUSUM feature during the fall events, and (c) fall detection accuracy results for a varying number of trees N . Use of US raw data or US CUSUM features as RFA inputs are summarized as well.

are marginal when using more than $N = 25$ trees. Comparing both cases, namely, choosing the input features k_f according to the zones z without optimizing f and using all the input features $k_f = 64$ but optimizing the feature groups f , it is apparent that the latter option provides higher accuracy. In the following, we will exploit the last case.

2) *US Sensor*: The fall activity results using both the raw data and the CUSUM feature are now presented for the US sensor. Fig. 9(a) shows an example of observed raw data during a subject fall. A person enters the monitored area, walks, and at 120 s loses his balance and falls down. The CUSUM feature $g_t^{(US)}$ is computed in Fig. 9(b): notice that the peaks clearly indicate the fall events. Both CUSUM and raw data can be used as an input to the RFA; therefore, Fig. 9(c) compares these two settings for a varying number of RF trees N . However, it is apparent that, if US raw data are employed as an input to the RFA method, it is possible to achieve an accuracy of 0.75 (for $N = 20$ trees), while the accuracy reaches 0.93 using the US CUSUM features. Using the CUSUM for raw data distillation is thus the preferred choice.

3) *Fused IR and US Sensors*: In what follows, we now discuss the benefits of feature fusion using both the IR sensor array and the US sensor in terms of fall detection accuracy. Fig. 10 shows the accuracy observed after the fusion of the CUSUM features $F(\hat{\Theta}_t) = \{g_{t,m}^{(IR)}; g_t^{(US)}\}$. In this example

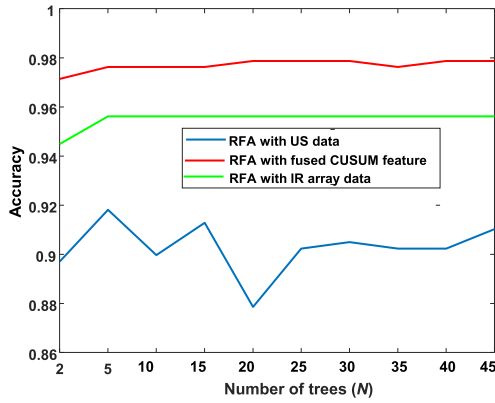


Fig. 10. Feature fusion and accuracy results using RFA versus the number of trees N in the zone $z = 5$ and the feature subsets size equal to $f = 15$.

TABLE I

COMPLEXITY AND ACCURACY VERSUS THE NUMBER OF TREES N AND VARIABLE NUMBER OF FEATURES f SELECTED BY RFA INCLUDING CUSUM PROCESSING AT EACH TREE T_i

feature subsets size f	# of trees N	Complexity \mathcal{O} [$\times 10^3$]	Accuracy
5	5	6.9	0.92
	20	16	0.94
	30	28	0.95
	50	68	0.95
10	5	7.4	0.9
	20	24	0.95
	30	43	0.95
	50	115	0.96
15	5	7.6	0.89
	20	29	0.93
	30	59	0.93
	50	151	0.95
20	5	85	0.94
	20	38	0.95
	30	75	0.95
	50	192	0.95

for the IR sensor, we use the pixels in the zone $z = 5$. Fall detection accuracy now is in the range of 0.94–0.98 while, in line with the previous analyses, the optimized number of RF trees is $N = 25$. While an appropriate pixel selection increases the accuracy, it reduces also the sensor FoV. Even if this drawback is tolerable for specific applications, it may not be acceptable for general-purpose scenarios. For this reason, in the following, we will not perform any pixel optimization. The computational cost/complexity of the proposed fall detection algorithm, including CUSUM feature calculation and RFA, is discussed in Section IV-B. To highlight the benefits of the proposed tool, a tradeoff analysis between accuracy and complexity is also considered with respect to conventional ML solutions.

B. Complexity Analysis

Computational complexity is discussed here through a comparative analysis between the proposed tool and conventional

TABLE II
ACCURACY, PROCESSING TIME FOR TRAINING AND INFERENCE, COMPLEXITY, AND MEMORY FOOTPRINT FOR LSTM, CNN, AND RFA

Algorithm	LSTM	CNN	RFA
Accuracy	0.98	0.9	0.94
Processing time (train./inf.) (s)	4.42/0.2	5.03/0.4	0.48/0.1
Complexity [$\times 10^3$]	3	13.9	1.12
Memory footprint (MB)	3.75	3.97	3.1

ML designs based on LSTM and CNN networks. LSTM and CNN layer structures are detailed in Fig. 11. The figure also summarizes the parameters used to evaluate the complexity for all algorithms. For a summary of all other parameters, see the related sections in the following. The proposed complexity analysis accounts for the CUSUM-based feature calculations as well as the ML-based classification tasks.

Considering the LSTM model [56], the complexity of one forward pass is ruled by the input layer size, the LSTM network edges, the FC neural network, and the output layer. Similarly, for CNN [55], the complexity depends on the input size, the 2-D convolutional layers, the FC, and the output layers. Finally, the RFA complexity [53] is determined by the input feature size, the number of trees, and the output layers. We will provide a detailed description of the parameters that rule the complexity for each case.

1) *Random Forest Algorithm*: It takes $k_f \leq M + 1 = 65$ features as inputs (from CUSUM) and chooses $f \leq k_f$ features for each tree T_i . The complexity of the proposed algorithm provides a measure of how many computing cycles (operations) the algorithm would take given an input of size k_f samples. According to [53] and [54], the complexity is defined as

$$\mathcal{O} \sim \underbrace{D \times f \times N}_{\text{RFA complexity}} + \underbrace{T \times k_f}_{\text{CUSUM processing}} \quad (12)$$

where N is the number of trees, D is the average depth, f is the feature number, and T is the number of samples for the CUSUM calculation as in (5) and (6). Notice that, at the training phase, the complexity should also consider the input feature k_f selection and it is calculated as $\mathcal{O} \sim k_f \times \log(k_f) \times D \times f \times N + T \times k_f$. The computational cost of the algorithm grows linearly with the number of the trees N , input k_f , and selected f features. By optimizing the number of features and trees, we could therefore reduce the complexity of the algorithm in exchange for an increase of the accuracy loss.

2) *Convolutional Neural Network*: The complexity is ruled by the 2-D convolution layer operations followed by the FC layer, in other words

$$\mathcal{O} \sim \underbrace{M \times N \times \left(\sum_{l=1}^{L_n} m_l \times n_l \times L_{f_l} \right)}_{\text{CNN}} \times N_f \times n_o + \underbrace{T \times k_f}_{\text{CUSUM}} \quad (13)$$

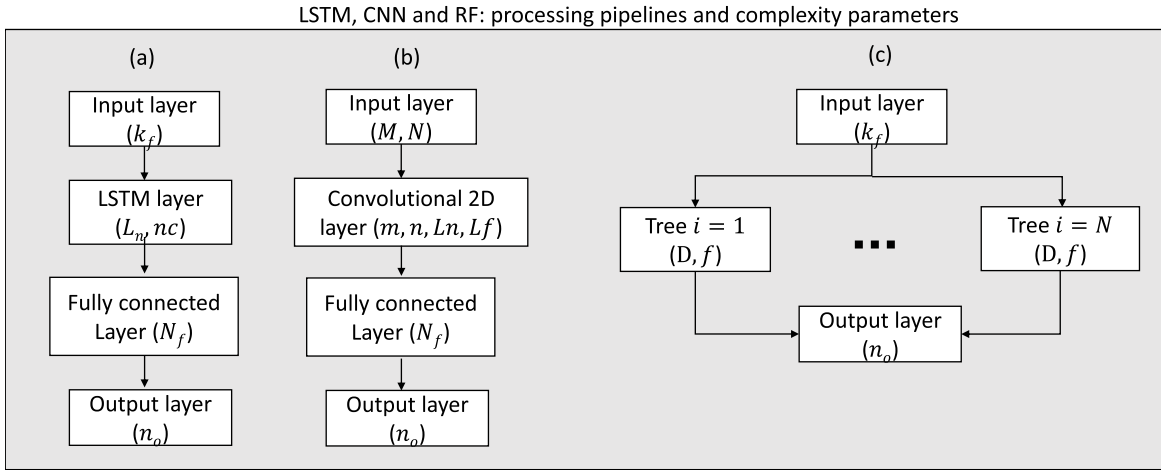


Fig. 11. Block diagrams and related parameters adopted for the complexity calculation for each layer of the compared algorithms: (a) LSTM. (b) CNN. (c) RFA.

where $M \times N$ is the input size, $m_l \times n_l$ is the filter size, L_{f_l} is the filter number per each layer, L_n is the CNN layer, N_f is the fully connected input units, and n_o is the number of outputs [55]. Similarly, as in (12), the last term $T \times k_f$ relates to the CUSUM calculation.

3) *Long Short-Term Memory*: The complexity of the LSTM algorithm is ruled by the total number of edges in the network (please refer to [56] for further details) and is defined as

$$\mathcal{O} \sim L_n \times (4n_c n_c + 4k_f n_c + n_o n_c + 3n_c) + N_f \times n_o + T \times k_f \quad (14)$$

where L_n is the number of LSTM layers, n_c is the memory cell, k_f is the input unit, N_f is the fully connected input, and n_o is the number of outputs. Notice that the computational time for a network with a moderate number of inputs is dominated by the term $L_n \times n_c \times (4k_f + n_o)$ since $k_f \gg n_c$. Therefore, the run time complexity can be simplified as

$$\mathcal{O} \sim \underbrace{L_n \times (n_c \times (4k_f + n_o))}_{\text{LSTM}} + \underbrace{N_f \times n_o + T \times k_f}_{\text{CUSUM}} \quad (15)$$

C. Complexity and Accuracy Tradeoff

The accuracy and computational complexity tradeoff is discussed here by assessing the proposed RFA method alone and also by a comparative analysis with conventional DL designs. Table I now focuses in more detail on the RFA algorithm. In particular, it analyzes the tradeoff between training complexity and accuracy for the variable number of trees N and the feature subsets size f selected for RFA processing at each tree \mathcal{T}_i .

In the cited table, we set the number of input features to $k_f = 65$. For all cases, the results confirm that increasing the number of trees and features almost produces an improvement in accuracy in exchange for increased complexity. Considering the hardware architecture of Fig. 5, balancing accuracy and computational cost is critical for practical implementation due to resource constraints. For example, setting an accuracy target of 0.96, the optimal number of the trees N and

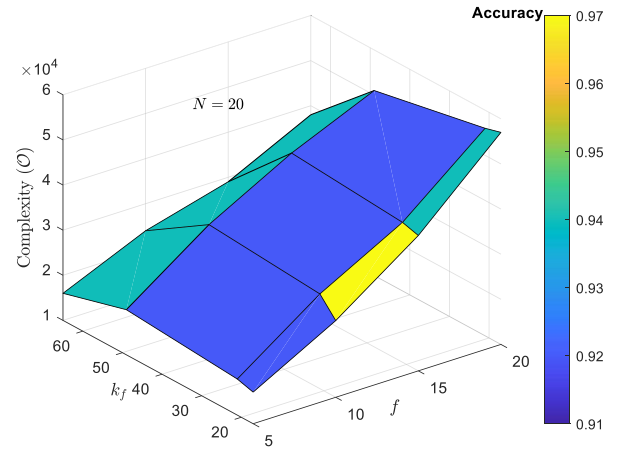


Fig. 12. Complexity and corresponding accuracy for a varying number of input features k_f and feature subset size f with $N = 20$.

features f combination is found to be equal to $f = 10$, $N = 50$, while the complexity is equal to 1.15×10^5 operations.

Similarly, by setting $N = 20$, Fig. 12 shows the relation between the number of input features k_f , the feature subsets size f selected for RFA, the computational complexity, and the corresponding accuracy. For example, for $N = 20$ and $k_f = 65$ features, the best choice for feature subsets size is $f = 5$ as this maximizes the accuracy up to 0.94.

In Table II, we compare the complexity results with LSTM and CNN using the complexity parameters in Fig. 11 with the corresponding values of Table III. CUSUM processing is excluded in complexity calculation to compare the ML tools' complexity. It is apparent that the RFA computational complexity is three times less than the LSTM one and 13 times less with respect to the CNN as well.

D. Training and Processing Time Analysis

In what follows, we analyze the processing time, the training time, and the memory usage of the proposed algorithms. Since

TABLE III
SUMMARY OF THE PARAMETERS VALUES ADOPTED FOR THE COMPARISON OF THE COMPUTATIONAL COMPLEXITY OF THE LSTM, CNN, AND RFA METHODS

Method	Parameters
LSTM	$k_f = 65, L_n = 1, n_c = 10, N_f = 2, n_o = 2$
CNN	$M = 9, N = 9, m = 4, 8, n = 4, 8, L_n = 2, L_f = 3, N_f = 2, n_o = 2$
RFA	$k_f = 65, D = 19, N = 20, f = 5, n_o = 2$

the actual processing time depends on the specific hardware and the software implementation, we will highlight relative comparisons. To simplify the analysis, computing times are measured using the same device, namely, a CPU @ 2.90 GHz and 16-GB RAM. RFA, CNN, and LSTM methods are compared, while the main results are summarized in Table II. According to the complexity analysis previously discussed, RFA has the lowest training and inference processing time. In particular, this is 50% less for inference than LSTM time and 75% less than the considered CNN algorithm. Regarding the training time, RFA uses ten times less time than CNN and LSTM. These results also confirm that a sensor fusion algorithm using RFA is a promising candidate for integration in the proposed multisensory edge device.

In Table II, we also quantify the memory usage for all the proposed methods. The goal is to compare the memory efficiency and the functionality of the proposed algorithm with conventional DL methods as integrated in an edge device with limited memory. Memory usage is quantified here as the size of memory used by each algorithm to produce a fall detection update. This is shown in megabytes (MB) in Table II. The result confirms that the proposed RFA method uses about 83% of memory (3.1 MB) compared with LSTM and about 78% with respect to the CNN algorithm.

V. CONCLUSION

In this article, we propose an integrated device that implements real-time feature-based fusion using data obtained from an IR array and a US sensor. The device implements data distillation by exploiting a change detection algorithm and extracts the features from both sensors using the CUSUM function. A low-complexity RFA is adopted for feature selection and processing. The algorithm has been validated using real data obtained from different field experiments to classify body fall and walk events in a smart space environment. The accuracy–complexity tradeoff is analyzed and compared with conventional ML approaches.

The results of the experimental analysis confirm that the proposed method is practical enough for implementation on resource-constrained edge devices with limited computational capacity and memory, specifically in the IoT applications. They demonstrated that the proposed approach can obtain an accuracy of 0.94, a reduction of the computational complexity of at least three times, 80% less processing time for the training, 50% less in the inference phase, and 10% less memory usage of the best ML competing method.

Although this article targeted IR and US sensors integration, the proposed method can be easily extended to integrate more

sensors since the proposed CUSUM feature processing is (raw) data agnostic.

Future works will consider the deployment of the proposed device in an industrial environment targeting a human–robot workspace multisensory ecosystem.

REFERENCES

- [1] D. K. Jain, Y. Li, M. J. Er, Q. Xin, D. Gupta, and K. Shankar, "Enabling unmanned aerial vehicle borne secure communication with classification framework for industry 5.0," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5477–5484, Aug. 2022.
- [2] D. Tola et al., "Towards easy robot system integration: Challenges and future directions," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Jan. 2022, pp. 77–82.
- [3] B. R. Ferrer et al., "Towards the adoption of cyber-physical systems of systems paradigm in smart manufacturing environments," in *Proc. IEEE 16th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2018, pp. 792–799.
- [4] S. Kianoush, S. Savazzi, M. Beschi, S. Sigg, and V. Rampa, "A multisensory edge-cloud platform for opportunistic radio sensing in cobot environments," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1154–1168, Jan. 2021.
- [5] J. Wang, D. Wang, S. Wang, W. Li, and K. Song, "Fault diagnosis of bearings based on multi-sensor information fusion and 2D convolutional neural network," *IEEE Access*, vol. 9, pp. 23717–23725, 2021.
- [6] S. Savazzi, S. Kianoush, V. Rampa, and U. Spagnolini, "Cellular data analytics for detection and discrimination of body movements," *IEEE Access*, vol. 6, pp. 51484–51499, 2018.
- [7] Y. Lu et al., "Outlook on human-centric manufacturing towards industry 5.0," *J. Manuf. Syst.*, vol. 62, pp. 612–627, Jan. 2022.
- [8] P. Fraga-Lamas, J. Varela-Barbeito, and T. M. Fernandez-Carames, "Next generation auto-identification and traceability technologies for industry 5.0: A methodology and practical use case for the shipbuilding industry," *IEEE Access*, vol. 9, pp. 140700–140730, 2021.
- [9] Y. Ge et al., "Contactless WiFi sensing and monitoring for future healthcare—Emerging trends, challenges and opportunities," *IEEE Rev. Biomed. Eng.*, early access, Mar. 7, 2022, doi: 10.1109/RBME.2022.3156810.
- [10] S. Kianoush, S. Savazzi, F. Vicentini, V. Rampa, and M. Giussani, "Device-free RF human body fall detection and localization in industrial workplaces," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 351–362, Apr. 2017.
- [11] H. Li et al., "Multisensor data fusion for human activities classification and fall detection," in *Proc. IEEE SENSORS*, Oct. 2017, pp. 1–3.
- [12] Q. Huang, C. Pan, and H. Liu, "A multi-sensor fusion algorithm for monitoring the health condition of conveyor belt in process industry," in *Proc. 3rd Int. Conf. Ind. Artif. Intell. (IAI)*, Nov. 2021, pp. 1–6.
- [13] H. Li, A. Mehul, J. Le Kerne, S. Z. Gurbuz, and F. Fioranelli, "Sequential human gait classification with distributed radar sensor fusion," *IEEE Sensors J.*, vol. 21, no. 6, pp. 7590–7603, Mar. 2021.
- [14] E. Molino-Minero-Re, A. A. Aguilera, R. F. Brena, and E. Garcia-Ceja, "Improved accuracy in predicting the best sensor fusion architecture for multiple domains," *Sensors*, vol. 21, no. 21, p. 7007, Oct. 2021.
- [15] M. G. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, "Machine learning at the network edge: A survey," *ACM Comput. Surv.*, vol. 54, no. 8, pp. 1–37, Nov. 2022.
- [16] W. Y. B. Lim et al., "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.
- [17] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

- [18] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.
- [19] J. Wang et al., "Bandwidth-efficient live video analytics for drones via edge computing," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2018, pp. 159–173.
- [20] G. Qian, S. Lu, D. Pan, H. Tang, Y. Liu, and Q. Wang, "Edge computing: A promising framework for real-time fault diagnosis and dynamic control of rotating machines using multi-sensor data," *IEEE Sensors J.*, vol. 19, no. 11, pp. 4211–4220, Jun. 2019.
- [21] B. Butler. (2017). *What Is Edge Computing and How It's Changing the Network*. Network World. Accessed: Mar. 7, 2018. [Online]. Available: <https://www.networkworld.com/article/3224893/internet-of-things/what-is-edge-computing-and-how-it-s-changing-the-network.html>
- [22] J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, C. Gan, and S. Han, "On-device training under 256 KB memory," 2022, *arXiv:2206.15472*.
- [23] F. Samie, L. Bauer, and J. Henkel, "Hierarchical classification for constrained IoT devices: A case study on human activity recognition," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8287–8295, Sep. 2020.
- [24] F. Samie, L. Bauer, and J. Henkel, "From cloud down to things: An overview of machine learning in Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4921–4934, Jun. 2019.
- [25] S. Mekruksavanich and A. Jitpattanakul, "Smartwatch-based human activity recognition using hybrid LSTM network," in *Proc. IEEE SENSORS*, Oct. 2020, pp. 1–4.
- [26] Y. Zhang, Q. Liu, Y. Wang, and G. Yu, "CSI-based location-independent human activity recognition using feature fusion," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.
- [27] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64270–64277, 2018.
- [28] S. Zheng et al., "Deep reinforcement learning based DNN model partition in edge computing-enabled metro optical network," in *Proc. Asia Commun. Photon. Conf. (ACP)*, Oct. 2021, pp. 1–3.
- [29] M. Basseville et al., *Detection of Abrupt Changes: Theory and Application*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.
- [30] J. Geng, B. Zhang, L. M. Huie, and L. Lai, "Online change detection of linear regression models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 4910–4914.
- [31] S. Kianoush, S. Savazzi, V. Rampa, L. Costa, and D. Tolochenko, "Calibration-free target detection based on thermal and distance sensor fusion," in *Proc. IEEE Sensors*, Oct. 2021, pp. 1–4.
- [32] Y. Zhang, H. Wang, and F. Xu, "Face detection based on multi task learning and multi layer feature fusion," in *Proc. 6th Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, Oct. 2017, pp. 289–293.
- [33] F. Zhou, X. Li, and Z. Wang, "Efficiently user-independent ultrasonic-based gesture recognition algorithm," in *Proc. IEEE SENSORS*, Oct. 2019, pp. 1–4.
- [34] S. Savazzi, V. Rampa, L. Costa, and D. Tolochenko, "Analysis of body-induced thermal signatures for social distancing monitoring," in *Proc. IEEE SENSORS*, Oct. 2020, pp. 1–4.
- [35] W. Zhang, X. Sui, G. Gu, Q. Chen, and H. Cao, "Infrared thermal imaging super-resolution via multiscale spatio-temporal feature fusion network," *IEEE Sensors J.*, vol. 21, no. 17, pp. 19176–19185, Sep. 2021.
- [36] P. Chodon et al., "Passive infrared (PIR) sensor based security system," *Int. J. Electr., Electron. Comput. Syst.*, vol. 14, no. 2, pp. 1–5, 2013.
- [37] M. Z. Saeed, R. R. Ahmed, O. B. Samin, and N. Ali, "IoT based smart security system using PIR and microwave sensors," in *Proc. 13th Int. Conf. Math., Actuarial Sci., Comput. Sci. Statist. (MACS)*, Dec. 2019, pp. 1–5.
- [38] A. D. Shetty, D. Padubidri, B. Shubha, and K. Suryanarayana, "Detection and tracking of a human using the infrared thermopile array sensor—'Grid-EYE,'" in *Proc. Int. Conf. Intell. Comput., Instrum. Control Technol. (ICICT)*, Jul. 2017, pp. 1490–1495.
- [39] S. Savazzi, V. Rampa, L. Costa, S. Kianoush, and D. Tolochenko, "Processing of body-induced thermal signatures for physical distancing and temperature screening," *IEEE Sensors J.*, vol. 21, no. 13, pp. 14168–14179, Jul. 2021.
- [40] L. Klibanov et al., "Preliminary analysis of TDK CH101 ultra-low power integrated ultrasonic time-of-flight range sensor," Ned. Maude, Todd and Rod Inc., Ottawa, ON, Canada, Tech. Rep. 346931359, 2022.
- [41] J. Jung, W. Lee, W. Kang, E. Shin, J. Ryu, and H. Choi, "Review of piezoelectric micromachined ultrasonic transducers and their applications," *J. Micromech. Microeng.*, vol. 27, no. 11, Nov. 2017, Art. no. 113001.
- [42] I. Guvenc and C.-C. Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 3, pp. 107–124, 3rd Quart., 2009.
- [43] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2017.
- [44] G. Flood, A. Heyden, and K. Åström, "Estimating uncertainty in time-difference and Doppler estimates," in *Proc. 7th Int. Conf. Pattern Recognit. Appl. Methods (ICPRAM)*, 2018, pp. 245–253.
- [45] A. Tuono, Y. Boev, and F. Kerschbaum, "Non-interactive private decision tree evaluation," in *Data and Applications Security and Privacy XXXIV (Lecture Notes in Computer Science)*, vol. 12122. Switzerland: Springer, 2020.
- [46] N. R. Nurwulan and G. Selamaj, "Random forest for human daily activity recognition," *J. Phys., Conf. Ser.*, vol. 1655, no. 1, Oct. 2020, Art. no. 012087.
- [47] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [48] M. Pal, "Random forest classifier for remote sensing classification," *Int. J. Remote Sens.*, vol. 26, no. 1, pp. 217–222, Feb. 2007.
- [49] V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, "An assessment of the effectiveness of a random forest classifier for land-cover classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 67, pp. 93–104, Jan. 2012.
- [50] A. Kaur and R. Malhotra, "Application of random forest in predicting fault-prone classes," in *Proc. Int. Conf. Adv. Comput. Theory Eng.*, Dec. 2008, pp. 37–43.
- [51] R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *J. Big Data*, vol. 7, no. 1, pp. 1–26, Dec. 2020.
- [52] T. K. Ho, "C4.5 decision forests," in *Proc. 14th Int. Conf. Pattern Recognit.*, vol. 1, 1998, pp. 545–549.
- [53] L. Štěpánek, F. Habarta, I. Malá, and L. Marek, "A random forest-based approach for survival curves comparing: Principles, computational aspects and asymptotic time complexity analysis," in *Proc. 16th Ann. Comput. Sci. Inf. Syst. (FedCSIS)*, Sep. 2021, pp. 301–311.
- [54] K. Hassine, A. Erbad, and R. Hamila, "Important complexity reduction of random forest in multi-classification problem," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 226–231.
- [55] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
- [56] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, Sep. 2014, pp. 338–342.



Sanaz Kianoush (Member, IEEE) received the Ph.D. degree in electronic engineering from the University of Pavia, Pavia, Italy, in 2014.

In 2014, she was a Postdoctoral Researcher with the Institute of Electronics, Computer and Telecommunication Engineering (IEIT), Consiglio Nazionale delle Ricerche (CNR), Milan, Italy. In 2022, she joined IEIT-CNR as a Researcher. In 2018, she was a Visiting Researcher with Aalto University, Espoo, Finland. From 2008 to 2010, she was a Lecturer

with Azad University, Tehran, Iran. Her research interests include statistical signal processing, sensor fusion in communication systems using machine learning and federated learning, design algorithms for device-free radio-based localization, and human activity recognition.

Dr. Kianoush received the IEIT-CNR Young Research Award (PreGio) in 2018.

Stefano Savazzi (Member, IEEE) received the M.Sc. and Ph.D. (Hons.) degrees in information and communication technology from the Politecnico di Milano, Milan, Italy, in 2004 and 2008, respectively.

In 2012, he joined the Institute of Electronics, Computer and Telecommunication Engineering (IEIT), Consiglio Nazionale delle Ricerche (CNR), Milan, as a Researcher. He was a Visiting Researcher with Uppsala University, Uppsala, Sweden, in 2005, and the University of California at San Diego, La Jolla, CA, USA, in 2007. He has co-authored over 100 scientific publications (Scopus). He is serving as the Principal Investigator for the Horizon EU project Holden. His current research interests include distributed signal processing, federated and distributed machine learning, and green networking aspects for the Internet of Things, including radio localization, RF holography, and vision technologies.

Dr. Savazzi received the Dimitris N. Chorafas Foundation Award in 2008. He is serving as an Associate Editor for *Frontiers in Communications and Networks*, *Sensors*, and *Wireless Communications and Mobile Computing*.



Vittorio Rampa (Member, IEEE) received the Dott. Ingegneria Elettronica (Hons.) degree from the Politecnico di Milano (POLIMI), Milan, Italy, in 1984.

In 1986, he joined the National Research Council (CNR), Milan, as a Researcher. From 1999 to 2014, he has been an Adjunct Professor at POLIMI, where he taught courses on software-defined radio algorithms and architectures, and radio communication and navigation systems. From 2001 to 2022,

he became a Senior Researcher at the Institute of Electronics, Computer, and Telecommunication Engineering (IEIT-CNR), where he is a Research Associate. His research interests include signal processing algorithms and architectures for wireless communications, radio vision techniques for people detection and localization, body models for radio passive sensing, and federated learning algorithms for cooperative and green applications.

Mr. Rampa has been a member of the Steering Committee GTTS-4 of the National Cluster Fabbrica Intelligence since 2018.

Leonardo Costa is an Electronic Engineer with signal processing and digital systems expertise. He is also the Research and Development Manager at Cognimade S.r.l., Melzo, Italy. His job experience includes software development for radars, embedded software development for different architectures, such as ARM Linux and Labview-based systems, firmware development on ARM Cortex-M microcontrollers and field-programmable gate array (FPGA), printed circuit board (PCB) design for the Internet of Things (IoT) devices, sensors, low-power radio-based products, industrial I/O, gateways, and data loggers. He also gained experience with different standards and rules, such as radio equipment device directive 2014/53/EU, ECSS software development life cycle, and electronic safety-related systems IEC 61508.

Denis Tolochenko is an Electronic Technician with printed circuit board (PCB) design and PCB testing expertise. He is also a Research and Development Expert at Cognimade S.r.l., Melzo, Italy. His job experience includes software development for radars and embedded software development for different architectures, such as ARM Linux. His job experience includes PCB design and testing for digital systems and radio-based systems, embedded software development for data loggers, and web-based Internet of Things (IoT) platforms.