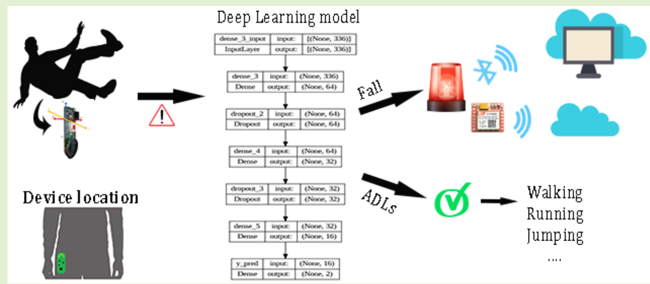
A Novel Embedded Deep Learning Wearable Sensor for Fall Detection

Sara Campanella, *Student Member, IEEE*, Alaa Alnasef, Laura Falaschetti, *Member, IEEE*, Alberto Belli, Paola Pierleoni, and Lorenzo Palma, *Member, IEEE*

***Abstract*—Falls and their aftermath pose significant healthcare challenges, impacting individuals across various age groups and occupational backgrounds. These incidents detrimentally affect functional mobility and overall quality of life, necessitating a comprehensive approach to fall detection systems in diverse populations. Therefore, wearable devices are necessary to continuously monitor activities. This work introduces a novel deep-learning model specifically optimized for edge devices capable of detecting falls. The wearable sensor integrates a pressure sensor, a three-axis gyroscope, and a three-axis accelerometer. The developed system works in real time with the dual objective of identifying the activities carried out and classifying them as falls or daily life activities. We evaluated this approach using both our self-collected dataset and a publicly available one (SisFall). Furthermore, in our dataset, we also introduced the syncope between falls that the sensor must be able to detect. Results demonstrate that while maintaining low cost, low complexity of the model, low-power consumption, and high-speed data processing, combining usage of the three sensors and deep learning (DL) algorithm allows to obtain an accuracy of 99.38% and an inference time of 25 ms.**



***Index Terms*—Activities of daily living (ADLs), deep learning (DL), edge computing, electronic devices, embedded systems, fall detection, occupational health, wearable sensors.**

I. Introduction

THE degeneration of the balance control system in the elderly, coupled with the increasing prevalence of age-related illnesses, has prompted extensive research. With a growing aging population and longer lifespans, preserving mobility is paramount. Researchers and physicians are actively investigating the functioning of the balance control system to assess its status [1]. Aging often involves a gradual decline in sensorimotor function, affecting the somatosensory, visual, and vestibular systems. Age-related changes in reaction times and stability limits lead to reduced balance control, particularly under cognitive challenges and unexpected postural disruptions [2]. Moreover, elderly individuals exhibit increased postural sway, which could eventually result in falls [3]. The World Health Organization (WHO) reports that roughly 28%–35% of those over 65 have a fall each year. When referring to older people above the age of 70, this ratio rises to 32%–42% [4]. According to projections, the number of people over 65 in the world will rise by 21.64% by the year 2050 [5]. Falls can result in wounds that cause discomfort, disabling conditions, and, in severe cases, early demise [6]. Besides the elderly, children, workers, and patients with visual problems also suffer from falls [7]. Unexpected falls among younger can result from a variety of conditions, including seizures, anemia, pregnancy, and sports. If problems are not identified and treated promptly, bone problems may arise [8]. In workplace environments, falls and fall-related injuries occur fairly frequently. In 2017, 887 workers died from falls, while 2 27 760 workers suffered injuries due to incidents that prevented them from working [9]. Nonfatal injuries not only can result in disability or functional impairment but also can increase the dread of falling again, which lowers an individual's autonomy and increases the social and economic burden [10]. Thus, early identification and intervention strategies to avoid workplace falls are critical to the economy, quality of life, productivity of the workplace, and health of workers.

Traditionally, dynamometric force platforms, which are tools for measuring the forces applied by a person on a support base, or motion capture techniques have been used for balance testing and evaluation [11], [12]. To properly understand the outcomes of these cutting-edge technologies, specialized tools and extensive knowledge are needed. They are not always

Manuscript received 1 February 2024; revised 4 March 2024; accepted 4 March 2024. Date of publication 18 March 2024; date of current version 1 May 2024. The associate editor coordinating the review of this article and approving it for publication was Prof. Ajay G. P. Kottapalli. *(Corresponding author: Lorenzo Palma.)*

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

The authors are with the Department of Information Engineering, Universitá Politecnica delle Marche, 60131 Ancona, Italy (e-mail: s.campanella@pm.univpm.it; s1107083@studenti.univpm.it; l.falaschetti@staff.univpm.it; a.belli@staff.univpm.it; p.pierleoni@staff.univpm.it; l.palma@staff.univpm.it).

Digital Object Identifier 10.1109/JSEN.2024.3375603

appropriate due to their high cost and need for specialized training before installation and data interpretation [13]. Body-wearable sensor technology has advanced significantly in recent years, relying on electromechanical sensors to precisely detect and track the body's movements and physical activity in unconstrained settings [14], [15], [16]. Gyroscopes and accelerometers are two examples of wearable sensors used by fall detection systems [17], [18]. These sensors are made to readily connect to the human body and have progressively gotten lighter and smaller. Specifically, an accelerometer and an angular velocity sensor combined to form a single inertial measurement unit (IMU) allow for low-cost, low-complexity, low-power, and high-speed data processing [19]. This makes the IMU popular for a variety of analyses, such as motion analysis and fall detection [20]. These devices can gather many signals, which can be utilized to distinguish between falls and activities of daily living (ADLs). ADLs include a broad range of behaviors that define people's habits, particularly in their homes, such as standing, sitting, and walking [21]. Periodic patterns in inertial signals, that correspond to the activity the subject is performing, are indicative of the ADL phase. An abrupt surge in the acceleration and angular velocity signals indicates the start of a fall, which is defined as the change from the ADL to the falling phase [22], [23].

For wearable technology, two main types of algorithms can be distinguished: threshold-based and artificial intelligence (AI)-based algorithms. Despite their excellent detection accuracy and low computational cost, threshold-based algorithms are particularly difficult to adapt for novel fall kinds and user characteristics [24], [25]. While AI techniques are considered more sophisticated approaches to address this problem, their effectiveness depends on large sample sizes, and there are currently limited datasets available to study these occurrences [26]. Moreover, given the constrained hardware capabilities of wearable devices, the challenge of creating highly accurate embedded algorithms at a reasonable computing cost remains unsolved [27]. A reliable and effective fall detection system must function well in several areas, such as accuracy, reaction time, and power consumption [28]. In literature, several works try to overcome some of the aforementioned limitations [29], [30], [31], but even though they obtain promising results, embedding the algorithm remains an open problem. Hence, we aim to develop a compact neural network suitable for operation on an embedded controller and to refine the model for practical real-world applications. Furthermore, we assessed the influence of integrating a pressure sensor signal and an additional type of fall, the syncope, into our self-collected dataset.

This article is organized as follows. Section II reports the data acquisition protocols and the preprocessing steps as well as the hardware and software implementation. In Sections III and IV, a deep discussion of the obtained results is reported, while in Section V, the pros and cons and future studies are presented.

A. Literature Review

The field of fall monitoring is still developing, with new systems being released regularly. There are several taxonomies on fall monitoring in the literature, but most of them are designed for systems that use ambient, video, and audio sensors for general monitoring. We offer a review of the literature that is especially focused on wearable sensor devices and the monitoring systems that use them, as will be discussed below.

Jain and Semwal [23] proposed a preimpact fall detection system that can detect falls within 0.5 s of the initiation phase, using the ensemble of convolutional neural networks (CNNs) and long short-term memory networks (LSTMs). The fusion of SisFall and Kfall datasets was used to train the model, which contains acceleration and gyroscope data of emulated falls. This study is limited by the fact that the databases include replicated falls. They are unsure if this model can identify pathological falls in the elderly and the feature extraction methodology needs further evaluation. Kabir et al. [32] proposed a class ensemble-based deep learning (DL) architecture integrated with CNN and LSTM, which was evaluated on SisFall and UMAFall using accelerometer and gyroscope data to detect nonfall, prefall, and fall classes. The results show high accuracy, but it comes with limitations, such as the low number of elderly subjects in the two datasets, the use of emulated falls under a controlled environment, and the high computational complexity of the model. Danilenka et al. [33] proposed a real-time fall detection system used for occupation health and safety using an accelerometer on a tag connected to a raspberry pie to detect (fall and ADL) classes, using an LSTM model. The authors reported that the performance suffered from high false positives (FPs), especially with activities with high acceleration, and a low sampling frequency of 2 Hz, which is lower than the recommended sampling frequency. Son et al. [34] focused on fall detection in farm workers. The researchers collected their dataset and compared it with the SisFall dataset. Two machine learning techniques, k-nearest neighbors (kNNs) and support vector machine (SVM), and an artificial neural network (ANN) were compared. The ANN showed the highest performance in binary classification, while SVM performed better in multiclass classification. With potential application in real-time monitoring, the results will be different in real-world scenarios, according to the researchers, and the inclusion of different age groups will have an impact on the motion recognition rates. In the work proposed by Vishnu et al. [35], they described a method for identifying human falls in surveillance footage by employing a fall motion mixture model (FMMM) to distinguish between falls and nonfall occurrences. The effectiveness of the suggested approach is shown using a range of surveillance video datasets with views from multiple cameras, wide-angle cameras, and narrow-angle cameras. Due to the presence of comparable visual cues, certain nonfall events are categorized as fall events. Therefore, some fall and nonfall events have subtle variations that the suggested method cannot handle. Al_Hassani and Atilla [36] focused on the long-standing issue of tracking moving turbulence to increase the accuracy of detection. They trained a four-stage forward neural network model, taking into account the fact that variables, such as the subject's bulk, speed, and gait pattern, can affect how accurately different actions are detected. They found that the overall prediction accuracy was 98.615%

when we combined the 4SFNN with the advanced method of particle swarm optimization. Choi et al. [31] proposed the use of a directed acyclic graph-convolutional neural network (DAG-CNN) for classifying falls, near falls, and ADLs. The researchers used a custom IMU containing an accelerometer and gyroscope, which was attached to the waist. A 40-feature vector was used as input to the model, and they found that using a combination of accelerometer and gyroscope data gives better performance than using a single sensor. The latency of the system was 58.5 ms. In addition, they noted certain drawbacks, like the fact that the subjects were young and healthy and that the falls were simulated in the lab; finally, the study neglected to consider the feasibility of implementing the system in real time. Nahian et al. [37] created a fall detection system that integrated various public datasets for training and testing and compared various machine learning algorithms. Moreover, feature reduction and extraction were used. For this binary classification challenge (falls versus ADLs), SMV was computed using a single accelerometer. When the suggested method was used and compared with the literature, the RF classifier yielded the best results. The method's shortcomings included the extraction of a large number of features and the lack of testing in real-world circumstances. A machine learning framework for fall detection and everyday activity recognition was developed by Chelli and Pätzold [38]. They identified seven different activities, including falls and daily activities, by using acceleration and angular velocity data from two public datasets. They did this by taking time- and frequency-domain features from the acceleration and rotational velocity data and using them to feed four different classification algorithms: ANN, ensemble bagged tree, quadratic SVM, and kNN. New features that improve the performance of the classifier are extracted from the power spectral density of the acceleration. Luna-Perejón et al. [26] proposed four different recurrent neural networks (RNNs) architectures using the SisFall dataset, and the models were deployed on two STM32 MCUs, for the prediction of three classes (BKG, alert, and fall). The findings demonstrated the feasibility of real-time usage of these types of RNN models (LSTM and GRU), with 35-ms inference time at 25-Hz sampling frequency, with relatively low power consumption, and an accuracy around 96%. Similarly, Musci et al. [39] utilized an extended version of the SisFall dataset, enriched with temporal annotations, to train the proposed fall detection model. The dataset augmentation involved marking actual fall occurrences using an annotation tool. The model employed a minimal RNN architecture based on LSTM cells, tailored for onboard wearable devices with limited resources. Implemented on the SensorTileR© board, the model achieved an accuracy exceeding 96% for real-time fall detection, showcasing its viability for practical applications. This research underscores the importance of appropriate datasets, efficient microcontrollers, and simplified DL architectures in advancing wearable device-based fall detection systems. Shahzad and Kim [40] discuss the development of an Android-based fall detection system called FallDroid; the algorithm employs a threshold-based method to effectively filter out most ADLs, then uses multiple kernel learning SVMs (MKL-SVM) for classifying fall-like events, and reduces false alarms, demonstrating promising results in terms of accuracy, sensitivity, and specificity. However, further research is suggested to validate the system's performance in real-world scenarios and explore potential enhancements in feature engineering. Using an LSTM model trained on the MobiACT dataset and running on a laptop, Ajerla et al. [41] proposed a fall detection system. The data stream for real-time predictions was collected from the MetaMotionR sensor and sent via Bluetooth to the laptop to get the predictions (fall–nonfall). The model will run only if the acceleration vector magnitude is above a predetermined threshold. Although the study's reported accuracy of 95.8% was less than that of previous research, the researchers claimed that it provided valuable information about the best sampling frequency of 50 Hz and the ideal location of the sensor, which is the waist.

The use of Wi-Fi channel state information (CSI) for motion detection is a persuasive method that guarantees accurate, economical, and covert surveillance [42], [43]. By utilizing wireless signals, this technique makes it possible to reliably detect movement without the use of physical sensors [44]. It is an attractive choice for many applications due to its precision and affordability [45]. Its nonintrusive nature makes it an ideal solution for safeguarding individuals, alerting caregivers or emergency services in the event of a fall and enhancing overall safety and well-being. For example, Abdelnasser et al. [46] created a system called WiGest that uses variations in Wi-Fi signal strength to detect hand gestures made in the vicinity of a user's mobile device. Similarly, Wang et al. [47] propose a human activity recognition and monitoring system (CARM) based on CSI. First, they put forth a CSI-speed model that measures the relationship between CSI dynamics and the speeds at which people move. Second, they provide a CSI-activity model that measures the correlation between activity levels and human movement velocities.

II. Materials and Methods

This part will address the tools used for data collection and decision-making, examine the dataset used to train the model, and then go through the developed model and the training strategies implemented.

A. Hardware

This study presents a wearable fall detection device that is wireless and integrated into a small printed circuit board (PCB), enabling the user to move freely.

The STM32U5 series of ultralow power microcontrollers, which includes the "STM32U575xx" microcontroller, was utilized. It can run at rates of up to 160 MHz and is based on the powerful Arm Cortex-M33 32-bit RISC core; however, in our application, we used an external 8-MHz crystal for power saving, and we used the CUBEMX clock tree configuration to set HCLK to 160 MHz. Our project's inertial module (LSM6DSOX), which has a 3-D digital gyroscope and accelerometer, is an essential part of motion tracking and fall detection. The accelerometer with ±16 g and gyroscope with ±2000 DPS resolution and 26-Hz sampling frequency was used.

To investigate how the pressure signal affected the fall detection algorithm and ascertain whether it may increase

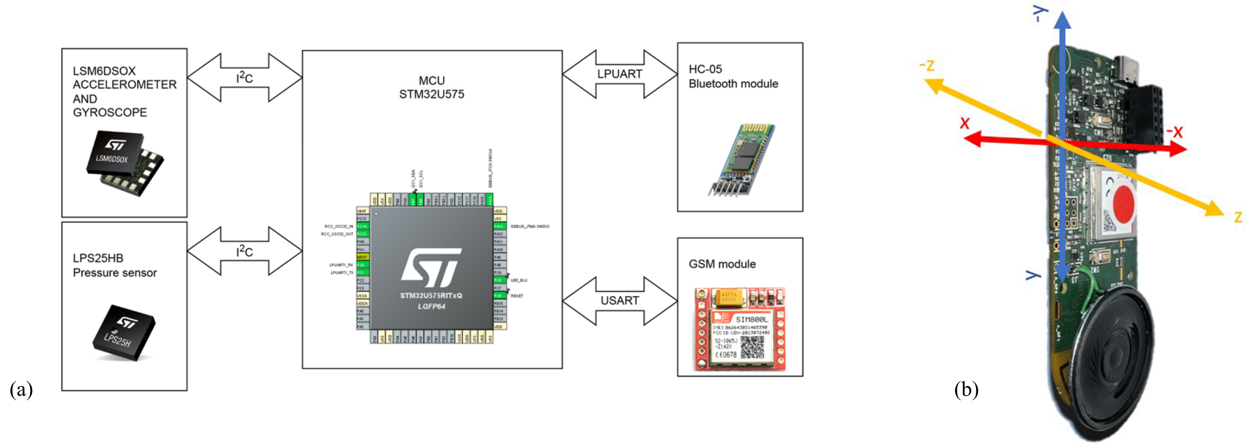


Fig. 1. (a) Block diagram of the wireless sensor node. (b) Final device used to record the data and its axis.

the model's accuracy in detecting falls, a pressure sensor (LPS25HB) was also utilized to track changes in the subject's height. It has a 16-sample moving average and 25-Hz sampling frequency.

Therefore, the self-developed embedded system made up of a microcontroller, 3-D accelerometer (±16 g, 0.488 mg/LSB, ODR26Hz), a gyroscope (±2000 dps, 70 mdps/LSB, ODR26Hz) (LSM6dSOX), and a pressure sensor (LPS25HB) (ODR25Hz, AVGT = 16, AVGP = 32, and 16-sample moving average). A 3.7-V 1150-mA Li-ion battery was used to power the circuit. Moreover, the device has a GSM module that can send data to a cloud infrastructure for informing the healthcare providers. In Fig. 1, all the aforementioned sensors and the final prototype are reported.

B. Firmware

The code for our STM32 microcontroller was developed using STM32CubeIDE [48]. Through the use of MATLAB application designer, we created our own MATLAB GUI, which allowed us to send data via Bluetooth and evaluate them for later use. We designed this GUI to transfer data in real time without the need for extra software, to perform algorithm analysis, and to obtain quick feedback on the functioning of the device. Furthermore, this configuration allowed us to test and validate the performance of the sensor in real time during the laboratory test.

Moreover, we have developed a C++ static library for fall detection on a custom STM32 board to streamline integration across diverse devices. The use of a static library ensures source code protection, with accessible functions encapsulated in a header file for effortless implementation. This approach enhances code reusability and eliminates the need for redundant coding. The library provides customizable parameters, including acceleration thresholds for DL model activation (AVM), which gives the user access to change the threshold at which the DL model works and yields the predictions. In addition, it exposes diagnostic data for efficient issue detection. The implementation leverages STM32CubeIDE for static library generation, complemented by a Cube.MX CMSIS-PACK library from Edge Impulse, housing our trained fall detection DL model.

C. Dataset

Three separate sets of data were used for the model's training and testing. These comprise the publicly accessible SisFall dataset [49] as well as a self-collected dataset that we obtained under controlled conditions and that contains additional sensor readings, such as the pressure sensor, that are absent from the SisFall dataset. Finally, we coupled our dataset with the SisFall dataset, or the combined dataset, to increase the model's accuracy. For the self-collected dataset, ethics review and approval were waived for this study, because the retrospective analysis of the recorded data was conducted using completely anonymous data. The experimental study did not involve any invasive or medical procedures and introduced no lifestyle changes. All subjects gave their informed consent before the collection and acquisition of the data, which was carried out in compliance with the ethical principles of the Helsinki Declaration.

A customized device, worn at the waist, fit with a gyroscope and two different kinds of accelerometers, was used to gather 15 different types of falls and a total of 19 different ADLs available in the SisFall dataset.

For the self-collected data, 18 ADLs and 16 fall types were done in our laboratory to simulate realistic circumstances that most frequently occur among older persons and workers. The candidates signed privacy questionnaires and got explanations regarding the protocol and the goal of the study. The activities chosen for this dataset are based on the SisFall dataset with the inclusion of a new fall type: syncope (also known as collapse or loss of consciousness [40]). Because syncope is associated with falls, occurs often in ER visits, and has a high death rate in the elderly, it is necessary to consider it and incorporate it into our fall detection system [50]. It was performed while leaning against a wall and then sliding along it ending up sitting.

The device was worn in the participants' right pockets without being secured. Two trials were performed with the sensor's z-axis pointing in the direction of movement and two trials with the z-axis pointing in the opposite direction.

In Table I, information about age, height, weight, and sex is reported for both datasets. The engagement of elderly subjects to perform our protocol was avoided, since it can cause injuries

TABLE I
AGE, HEIGHT, AND WEIGHT OF THE PARTICIPANTS IN THE SISFALL AND SELF-COLLECTED DATASETS

Dataset	Subjects	Sex	Age	Height (m)	Weight (kg)
SisFall	Elderly	Female	62-75	1.50-1.69	50-72
		Male	60-71	1.63-1.71	56-102
	Adult	Female	19-30	1.49-1.69	42-63
		Male	19-30	1.65-1.83	58-81
Self-collected	Adult	Female	24-34	1.68-1.72	65-68
		Male	24-40	1.58-1.83	50-90

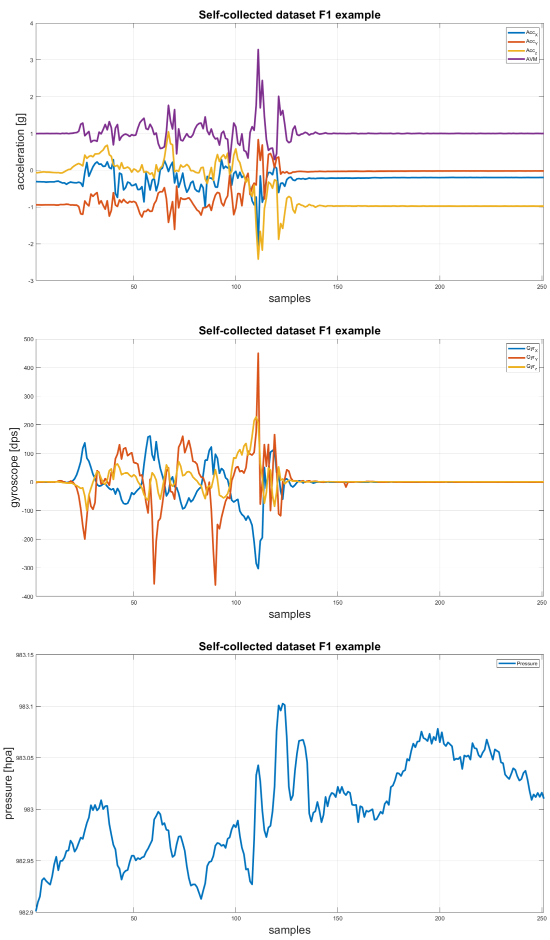


Fig. 2. Fall samples from self-collected data showing accelerometer, gyroscope, and pressure signals during a fall forward while walking caused by a slip.

and was not appropriate. All the experimental protocols used in the literature for the validation of fall detection systems employ young persons who simulate falls or a few elderly subjects performing the falls in a controlled environment [51]. Figs. 2 and 3 are reported self-collected accelerometer data representations and examples of the activities recorded, respectively.

D. Data Preprocessing

Three datasets were utilized to train and test the model's capacity to discriminate between ADL and fall classes, as previously stated. It is critical to select the window width and sample frequency of the collected data that will be fed to the model to maximize the accuracy in prediction. The SisFall dataset contains 3-D accelerometer and gyroscope data. The acceleration vector magnitude (AVM) was calculated for each trial in the SisFall dataset to improve the model and provide a new signal that is orientation-independent. After that, since the SisFall dataset contains different time windows for each task, sampled at 200 Hz, we downsampled to 25 Hz to match our sensor configuration, and a 12-s window size was set for all tasks. Therefore, we removed any signal that has a length of fewer than 5 s, since it does not contain important information (i.e., if a signal is 25-s long, the signal is divided into three windows 12-12-1 s, and the 1-s leftover signal is removed).

Similarly, our self-developed sensor was used to gather the acceleration, gyroscope, AVM, and pressure signals. Specifically, in terms of the internal sensor configurations, the sampling frequencies for the accelerometer and gyroscope were at 26 Hz, while the pressure sensor was set to sample data at 25 Hz. Therefore, we used a sampling frequency of 25 Hz and a window width of 10 s, which was adequate to record the fall event.

After this data manipulation, the total length was different for each of the three datasets used to train the model: 1) SisFall dataset: 20 h 12 m 25 s; 2) self-collected dataset: 6 h 26 m 0 s; and 3) combined dataset: 26 h 38 m 35 s.

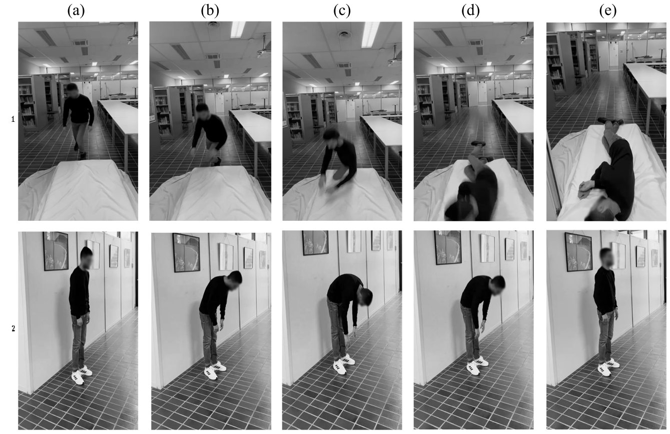


Fig. 3. Examples of activities recorded. (a1)-(e1): lateral fall while walking caused by a slip. (a2)-(e2): standing, slowly bending without bending knees, and getting up.

E. Features Extraction

The feature extraction and subsequent model implementation have been made on the Edge Impulse platform [52], a cloud-based machine learning operation (MLOps) that allows developers to create embedded and edge machine learning (TinyML) systems that can be implemented on a variety of hardware platforms [52]. A time-series data block with a 10 000-ms window size, a 2000-ms window increment, and Fs = 25 Hz was utilized as input. A sliding window covering all data lengths was implemented with an increase of 2000 ms if the data window exceeds 10 000 ms. We applied zero padding if the signals were less than 10 000 ms.

After obtaining the time-series data, the spectral analysis block uses it to compute features that are subsequently fed

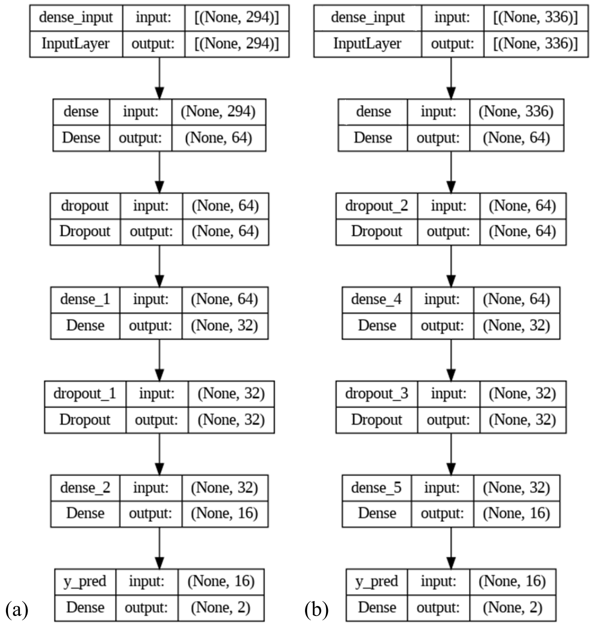


Fig. 4. (a) Model summary for SisFall and combined datasets. (b) Model summary for the self-collected dataset.

into the model. Our objective is to use spectrum analysis to extract information related to frequency that can be used to distinguish falls from ADLs. Because some movement patterns connected to falls may have different frequency characteristics than ordinary ADLs, this technique is particularly useful [53]. We employ Fast Fourier transformation—more specifically, 32 FFT length—for the spectral analysis. From this spectrum analysis, we derive a set of features for the Accx, Accy, Accz, Gyrx, Gyry, Gyrz, and AVM axes; in addition, we incorporate the pressure signal for our self-collected dataset. A classifier block will receive the features generated by the spectral analysis block. In Table II, all the 42 features are listed. After that, data were divided into a training set of 80% and a testing set of 20%. The training dataset was further divided into a 20% validation set.

F. Model

We selected the feedforward neural network (FFNN) as the model architecture, because it has a small memory footprint. The configuration is reported in Fig. 4.

The model consists of dropout layers to avoid overfitting and dense layers with the ReLU activation function and a softmax function for the last layer. The Adam optimizer and categorical cross-entropy loss have been chosen as hyperparameters. With dropout layers to improve resilience and the $l1$ regularization approach, this design is adaptable and modular. We tested two slightly different neural networks due to the presence of the pressure signal that forced us to change the input layer. Finally, we select 200 epochs with a learning rate of 0.0005.

The output of the classifier is either an "ADL" or a "fall" class.

G. Fall Detection Algorithm

In this section, the implemented fall detection system is presented. Fig. 5 reports an illustration of the fall detection system's flowchart. Initially, sensor data are continuously collected. This data collection persists until the AVM signal surpasses a predefined threshold of 2.5 g [54]. Only after the

TABLE II
SPECTRAL FEATURES

Num	Features	Num	Features
1	Spectral Skewness	22	Spectral Power 4.3 - 5.08 Hz
2	Spectral Power 0.04 - 0.12 Hz	23	Spectral Power 0.51 - 0.59 Hz
3	Spectral Power 2.73 - 3.52 Hz	24	Spectral Power 1.17 - 1.95 Hz
4	Spectral Power 0.74 - 0.82 Hz	25	Spectral Power 0.27 - 0.35 Hz
5	Spectral Power 0.82 - 0.9 Hz	26	Spectral Skewness LF
6	Spectral Kurtosis LF	27	Skewness
7	Spectral Power 0.39 - 1.17 Hz	28	Spectral Power 0.9 - 0.98 Hz
8	Skewness LF	29	Spectral Power 0.43 - 0.51 Hz
9	Spectral Power 3.52 - 4.3 Hz	30	Spectral Kurtosis
10	Spectral Power 0.98 - 1.05 Hz	31	Kurtosis
11	Spectral Power 0.59 - 0.66 Hz	32	RMS
12	Spectral Power 1.95 - 2.73 Hz	33	Spectral Power 5.08 - 5.86 Hz
13	Spectral Power 0.2 - 0.27 Hz	34	Spectral Power 7.42 - 8.2 Hz
14	RMS LF	35	Spectral Power 8.2 - 8.98 Hz
15	Spectral Kurtosis LF	36	Spectral Power 5.86 - 6.64 Hz
16	Spectral Power 0.35 - 0.43 Hz	37	Spectral Power 12.11 - 12.89 Hz
17	Spectral Power 1.13 - 1.21 Hz	38	Spectral Power 10.55 - 11.33 Hz
18	Spectral Power 1.21 - 1.29 Hz	39	Spectral Power 6.64 - 7.42 Hz
19	Spectral Power 1.95 - 2.73 Hz	40	Spectral Power 8.98 - 9.77 Hz
20	Spectral Power 1.05 - 1.13 Hz	41	Spectral Power 0.66 - 0.74 Hz
21	Spectral Power 0.12 - 0.2 Hz	42	Spectral Power 9.77 - 10.55 Hz

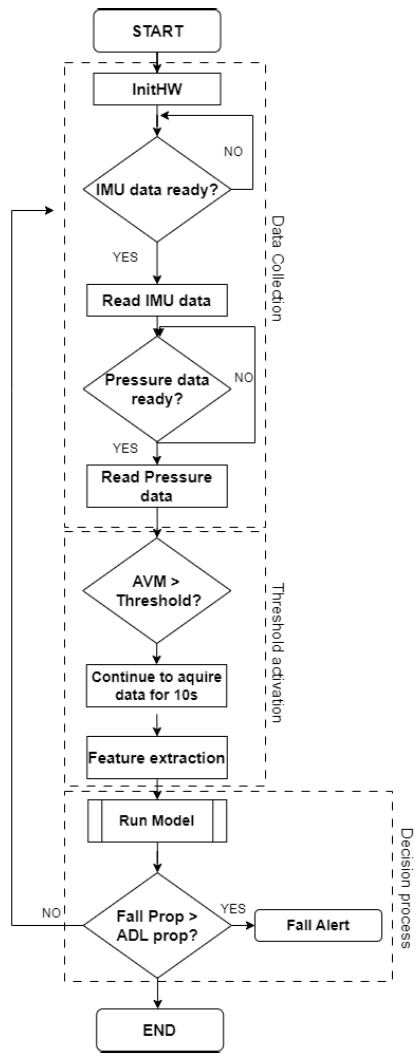


Fig. 5. Flowchart of the proposed firmware for our custom device.

threshold is exceeded, the DL algorithm will start working. In this way, we can preserve the battery life and reduce the computational burden. At this point, the probability of a fall event is identified (AVM > AVMth). From the sample in which the signal amplitude exceeds the threshold, data collection continues until we collect 5 s of data (10 s—250 samples in total). Notably, the code is designed to ensure that the fall event is centered within the recorded window and aims to enhance model predictions by aligning the window with the training data. Subsequently, the recorded window undergoes feature extraction. These extracted features serve as inputs for the model, which generates probabilities for both falls and ADLs. If the falling probability surpasses the ADL probability, a fall alarm is triggered. Employing the Bluetooth and/or the GSM module, the alarm can be sent to healthcare facilities or a cloud system.

III. Results

This section presents the experimental analysis in detail, including the evaluation metrics used to assess the models' performance. We also thoroughly examine the experimental data to offer insights into the efficacy of the suggested approach, the impact of various factors on its performance, and its practical implications. To evaluate the performance of the proposed method, we used accuracy and $F1$ score for each class. We also reported the true positive (TP), true negative (TN), FP, and false negative (FN) percentages. Moreover, to evaluate the performance of our devices, we computed the inference time and tested the battery consumption. Table III reports the results for all the different combinations of the dataset used for the training and testing phases.

Training the model with our self-collected dataset with and without the pressure signal and with the syncope demonstrates high accuracy and low values of loss [Fig. 6(a)]. Specifically, evaluating the model with our dataset without pressure shows promising results even though we did not consider elderly subjects. Both the TP rate and the model accuracy attain high percentages (98.05%). However, the FN rate is not very accurate, which might be problematic, because it means that a fall may go unnoticed even if it does occur. Adding the pressure signal (1859 windows for training and 457 for testing) helped in increasing the final accuracy as well as the TP and lowering the FN values. This is a positive aspect, since, to quickly respond and prevent health issues, it is essential to accurately identify falls and reduce the error rate. This implies that the robustness of the model can be greatly increased by adding a pressure sensor and, consequently, interacting with the different signals and sensors.

The most promising results were obtained when combining our self-collected dataset (without pressure) with the original SisFall dataset. We obtained a training accuracy of 99.9% with a loss of 0.01 on the validation set [Fig. 6(b)] and a testing accuracy of 99.38%. Moreover, it exhibited the lowest FN rate, making it the most effective dataset for fall detection. This can be because the training data were substantially increased to 6693 windows with a length of 5–12 s each, and we merged our dataset with the SisFall where elderly subjects' data are reported.

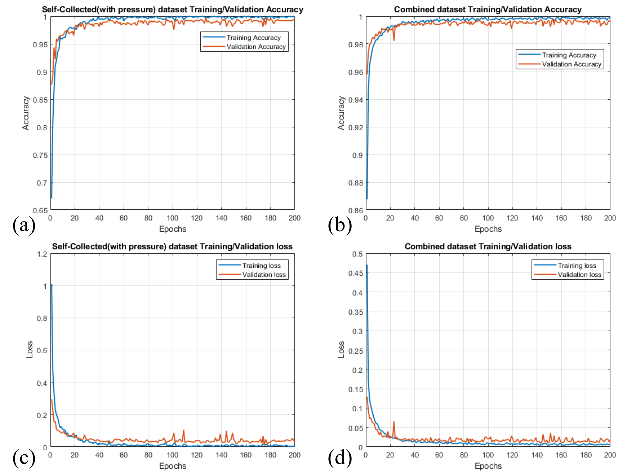


Fig. 6. Training and validation accuracy: (a) self-collected data and (b) combined dataset. Training and validation loss: (c) self-collected data and (d) combined dataset.

When the model was trained and tested using only the SisFall dataset, the testing accuracy was 99.3%. When the model was trained and tested using additional datasets, similar promising results were obtained (94.86%). This shows that our method is adaptable and capable of handling various data typologies and falls, producing solid results. The increase in the FP/FN rate can be the result of the various sensor sites where the data were recorded (pocket versus waist). All the proposed dataset combinations have an inference time of 25 ms, further validating our statements.

We tested the device using three different threshold values (1.5, 2.0, and 2.5 g) to determine how long the battery would last. We concurrently put all three devices in the same spot where we acquired the protocol and wore them for several days to verify the battery life in a real-world setting. Finally, we averaged the three devices' battery lives that were worn by each of the subjects involved in the protocol. The results demonstrate that lowering the activation threshold resulted in half of the average battery life (Table IV). From this, we can say that it is necessary to set the threshold based on the subject who is using the device. For more active individuals, such as workers or young people, the ADLs acceleration is higher and leads to the continuous overcoming of the threshold, so a higher baseline may be appropriate to prevent the algorithm from being activated unnecessarily. Conversely, a lower activation threshold can be used for the elderly, because they would wear the sensor in everyday activities where it is important to accurately differentiate falls from ADLs and act promptly.

IV. Discussion

In this study, we proposed a DL algorithm for fall detection embedded into a self-made wearable device. It is capable of distinguishing between the different ADLs and falls. Our experiments demonstrated that the proposed method achieved better performance than state-of-the-art approaches (Table V). The results show that our method can effectively enhance classification accuracy by learning the features specific to each class. Moreover, we developed a simple model architecture with a small memory footprint able to reach high accuracy compared to sophisticated approaches with higher inference

TABLE III
RESULTS TABLE FOR ALL THE TRAINING AND TESTING CONFIGURATIONS

Training data	Testing data	Testing Acc(%)	TP(%)	TN(%)	FP(%)	FN(%)	F1_ADL	F1_FALL	Inference (ms)	Peak RAM (Kbyte)	Flash Usage (Kbyte)
SisFall	SisFall	99.3	99.3	99.3	0.7	0.7	0.99	0.99	25	2.8	94.6
SisFall	self-collected (without pressure)	94.86	97.3	99.9	2.3	6.7	0.95	0.95	25	2.8	94.6
self-collected (with pressure)	self-collected (with pressure)	99.12	99.1	99.2	0.9	0.4	0.99	0.99	25	3	105.1
self-collected (without pressure)	self-collected (without pressure)	98.05	98.1	98.0	1.9	2.0	0.98	0.98	25	2.8	94.6
Combined dataset	Combined dataset	99.38	98.5	99.8	1.3	0.2	1	0.99	25	2.8	94.6

TABLE IV
BATTERY DURATION FOR EACH TESTED THRESHOLD

Threshold	Battery duration
1.5 g	28h 30m
2.0 g	29h 00m
2.5 g	53h 30m

time. In fact, we have experimented with both CNN and LSTM for our work. However, these networks demand substantial flash memory. CNN uses 1.5 GB; however, our method just uses 60 kb. This gets limited when you take into account the 2-MB memory of our microcontroller, since we have to sum to the 1.5 MB of the network other code for functionality. Despite attempting quantization to shrink the CNN network size, it adversely affected accuracy, dropping to around 70% with high FNs and positives. In addition to memory concerns, our approach boasts a 30-ms inference time, significantly faster than CNN's 500 ms or more. An inherent limitation of CNN lies in its inability to adapt to varying device orientations and sensor axes. While solutions like signal derivation exist, they compromise memory efficiency and computational simplicity. While solutions like deriving new signals for axis-independent training in CNN exist, our priority is minimal memory usage, swift inference, and computational simplicity. Hence, we adopted a straightforward approach—extracting features and feeding them into our simple architecture, yielding accurate results.

It is noteworthy that it demonstrates its capacity to handle unseen data, opening up new avenues for the analysis of novel activities. The network's performance holds up well when comparing the outcomes of training it with a single dataset and testing it with other data. This is especially significant, because we added a new kind of fall.

It is crucial to note that both TN and TP have high values. This underscores the model's strength and its precision in distinguishing between negative and positive cases, with a special emphasis on the significance of TP. Having a substantial number of TPs and a minimal count of FN is pivotal in effectively discerning between falls and nonfalls. This is especially important for TP, since there can be serious health effects for subjects if a fall is not detected, which emphasizes the fundamental necessity for accurate detection. We determined a strategic activation threshold to activate the DL model only if the 2.5-g value was overcome. This strategy has proven essential for preserving batteries and ensuring a more resource-conscious usage, guaranteeing efficient utilization. The model can save energy without sacrificing its ability to react quickly when necessary. Indeed, we put the aforementioned devices through a real-world test using 1.5-, 2.0-, and 2.5-g thresholds. The 1.5-g device's battery died after 28 h and 30 min and similarly did the 2.0-g device (29 h). In contrast, the 2.5-g device exhibited remarkable endurance, lasting 53 h and 30 min. This highlights the significant impact of employing the 2.5-g threshold, effectively doubling the device lifespan while selectively activating the DL model in practical scenarios.

Compared with the common architectures used in the literature, our FFNN architecture is unique. Musci et al. [39] used LSTM, a kind of RNN that can record long-term dependencies and is, hence, well suited for sequential input. Shahzad and Kim [40] combined the strength of SVM with the power of kernel approaches by utilizing an MKL-SVM. Gated recurrent unit (GRU), an LSTM variation renowned for its computational efficiency, was the choice made by Luna-Perejón et al. [26]. Our FFNN model, on the other hand, has a straightforward and effective architecture devoid of recurrent connections. Our approach is computationally lightweight due to the lack of recurrent loops, which allows for faster execution on microcontrollers with limited resources. Because of its straightforward architecture, which guarantees quicker inference times, it is ideally suited for real-time applications where timely fall detection is essential. Moreover, our model can minimize computational power usage. This inherent efficiency facilitates swift execution on microcontrollers, ensuring real-time fall detection. The absence of recurrent connections reduces memory requirements, enabling seamless integration into resource-constrained environments.

Using spectral analysis, our feature extraction approach extracts complex frequency patterns from accelerometer and gyroscope data. This method naturally accommodates the various and nuanced movement patterns linked to falls, strengthening the model's capacity for discrimination. By using the temporal patterns that LSTM and GRU networks learned, Luna-Perejón et al. [26] and Musci et al. [39] mostly depended on raw sensor data. With an emphasis on accelerometer data, Shahzad and Kim [40] combined threshold-based approaches with pattern recognition techniques.

In addition, in terms of performance metrics, our FFNN model achieves a testing accuracy of 99.38%. This surpasses previous results: Musci et al. [39] achieved 93.52%, Shahzad and Kim [40] reached 97.81%, and Luna-Perejón et al. [26] had 96.7%. Our model's superior accuracy underscores its precision in distinguishing falls from nonfall activities. Our model reaches a sensitivity of 99.79% and a specificity of 98.62%. High sensitivity ensures accurate fall detection for timely intervention, while specificity minimizes false alarms. Finally, our model has a speedy inference time of 25 ms, outpacing [26].

TABLE V
COMPARISON BETWEEN OUR BEST MODEL (COMBINED DATASET) AND THE LITERATURE

Study	Model	MCU Implementation	Testing Acc(%)	Sensitivity (%)	Specificity (%)	Precision (%)	Inference time(ms)
Our approach	FFNN	Yes	99.38	99.79	98.62	98.69	25
Musci et al. [39]	LSTM	Yes	93.52	90.43	96.6	/	/
Shahzad et al. [40]	MKL-SVM	Yes (smart phone)	97.81	99.52	95.19	/	/
Luna-Perejón et al. [26]	GRU	Yes	96.7	87.5	96.8	68.1	34

A. Limitations

There are some limitations in our work. First of all, protocol recording turned out to be a time-consuming process, and participants frequently showed signs of exhaustion afterward. There could have been differences in the data collected if their fatigue had affected their fall. The authenticity of the falls that were simulated during the recording sessions could be affected by such weariness. Moreover, the dataset we designated as "self-collected" was limited to a younger population, with no representation from older participants. Therefore, we decided to integrate the available data with the large SisFall dataset, which includes a significant number of falls from elderly people. By incorporating a more extensive and diverse dataset, the generalizability and reliability of the findings can be enhanced, leading to more robust conclusions. Furthermore, a fixed threshold is incorporated into our work (even though it can be changed by software); therefore, an automatic threshold is required to be able to adapt to different categories of subjects, such as workers, elderly, and so on.

V. CONCLUSION

Our study focused on evaluating a fall detection system, emphasizing the importance of a simple yet effective model architecture suitable for embedded systems. We designed our model with dense layers and a small memory footprint, ensuring its feasibility for real-world applications. Using a sampling frequency of 25 Hz, we enabled low-power data collection while maintaining accurate fall-detection capabilities. In addition, we integrated a custom library that allows for system portability across various sensors. This enhances compatibility and ensures that the system can seamlessly adapt to different environments, providing a flexible and robust solution. Despite its simplicity, our model exhibited exceptional accuracy and robustness with fast inference times compared with complex approaches present in the literature. This emphasizes the effectiveness of streamlined architectures, especially in resource-constrained environments. Including pressure signals and the syncope fall, improved model accuracy, underlining the significance of diverse sensors and sensor fusion in improving the model performance. Addressing the limitations of the existing datasets, especially in involving elderly subjects, is a collective challenge for the research community. Exploring various sampling frequencies and time windows remains a promising avenue for fine-tuning fall detection models. In the future, experimenting with different neural networks and algorithms could be a viable approach. The main goal would be to optimize the size of the network to make it compatible with microcontrollers. Investigating different setups or equipment, such as Wi-Fi CSI, may be helpful. It might also be thought about improving categorization by switching from binary to multiclass, which would be useful for preventive actions in eldercare as well as professional settings. The deployment of our proposed device has the potential to significantly enhance the safety and well-being of individuals at risk of falling as well as workers in dangerous environments, making a substantial impact on public health and society.

ACKNOWLEDGMENT

The authors wish to thank Esse-Ti S.r.l. for support in hardware development.

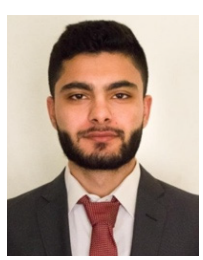
REFERENCES

[1] D. Winter, "Human balance and posture control during standing and walking," *Gait Posture*, vol. 3, no. 4, pp. 193–214, Dec. 1995.
[2] A. Zampogna et al., "Fifteen years of wireless sensors for balance assessment in neurological disorders," *Sensors*, vol. 20, no. 11, p. 3247, Jun. 2020.
[3] C. A. Laughton et al., "Aging, muscle activity, and balance control: Physiologic changes associated with balance impairment," *Gait Posture*, vol. 18, no. 2, pp. 101–108, Oct. 2003.
[4] W. H. O. Ageing and L. C. Unit, *WHO Global Report on Falls Prevention in Older Age*. Geneva, Switzerland: World Health Organization, 2008.
[5] L. Xing et al., "Falls caused by balance disorders in the elderly with multiple systems involved: Pathogenic mechanisms and treatment strategies," *Frontiers Neurol.*, vol. 14, Feb. 2023, Art. no. 1128092.
[6] A. Singh, S. U. Rehman, S. Yongchareon, and P. H. J. Chong, "Sensor technologies for fall detection systems: A review," *IEEE Sensors J.*, vol. 20, no. 13, pp. 6889–6919, Jul. 2020.
[7] S. Rastogi and J. Singh, "A systematic review on machine learning for fall detection system," *Comput. Intell.*, vol. 37, no. 2, pp. 951–974, May 2021.
[8] F. Shu and J. Shu, "An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost Android box," *Sci. Rep.*, vol. 11, no. 1, p. 2471, Jan. 2021.
[9] S. N. K. Kodithuwakku Arachchige, H. Chander, A. C. Knight, R. F. Burch V, and D. W. Carruth, "Occupational falls: Interventions for fall detection, prevention and safety promotion," *Theor. Issues Ergonom. Sci.*, vol. 22, no. 5, pp. 603–618, Sep. 2021.
[10] A. Hayat and M. Shan, "Fall detection system for labour safety," in *Proc. Int. Conf. Eng., Appl. Sci., Technol. (ICEAST)*, Jul. 2018, pp. 1–4.
[11] R. Romijnders, E. Warmerdam, C. Hansen, J. Welzel, G. Schmidt, and W. Maetzler, "Validation of IMU-based gait event detection during curved walking and turning in older adults and Parkinson's disease patients," *J. NeuroEng. Rehabil.*, vol. 18, no. 1, p. 28, Dec. 2021.
[12] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Comput. Methods Programs Biomed.*, vol. 117, no. 3, pp. 489–501, Dec. 2014.
[13] L. Abou et al., "Gait and balance assessments using smartphone applications in Parkinson's disease: A systematic review," *J. Med. Syst.*, vol. 45, no. 9, pp. 1–20, Sep. 2021.
[14] B. Najafi, D. Horn, S. Marclay, R. T. Crews, S. Wu, and J. S. Wrobel, "Assessing postural control and postural control strategy in diabetes patients using innovative and wearable technology," *J. Diabetes Sci. Technol.*, vol. 4, no. 4, pp. 780–791, Jul. 2010.
[15] H. Vasoya, H. Bhattasana, and R. G. Mishra, "A review of elderly fall detection systems using artificial intelligence," in *Proc. 7th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, May 2023, pp. 541–546.

[16] Z. Wang, V. Ramamoorthy, U. Gal, and A. Guez, "Possible life saver: A review on human fall detection technology," *Robotics*, vol. 9, no. 3, p. 55, Jul. 2020.
[17] P. Pierleoni et al., "A wearable fall detector for elderly people based on AHRS and barometric sensor," *IEEE Sensors J.*, vol. 16, no. 17, pp. 6733–6744, Sep. 2016, doi: 10.1109/JSEN.2016.2585667.
[18] P. Bet, P. C. Castro, and M. A. Ponti, "Fall detection and fall risk assessment in older person using wearable sensors: A systematic review," *Int. J. Med. Informat.*, vol. 130, Oct. 2019, Art. no. 103946.
[19] M. H. Rahmani, R. Berkvens, and M. Weyn, "Chest-Worn inertial sensors: A survey of applications and methods," *Sensors*, vol. 21, no. 8, p. 2875, Apr. 2021.
[20] T. H. Kim, A. Choi, H. M. Heo, K. Kim, K. Lee, and J. H. Mun, "Machine learning-based pre-impact fall detection model to discriminate various types of fall," *J. Biomech. Eng.*, vol. 141, no. 8, Aug. 2019, Art. no. 081010.
[21] P. Vallabh and R. Malekian, "Fall detection monitoring systems: A comprehensive review," *J. Ambient Intell. Hum. Comput.*, vol. 9, no. 6, pp. 1809–1833, Nov. 2018.
[22] R. Jain, V. B. Semwal, and P. Kaushik, "Stride segmentation of inertial sensor data using statistical methods for different walking activities," *Robotica*, vol. 40, no. 8, pp. 2567–2580, Aug. 2022, doi: 10.1017/S026357472100179X.
[23] R. Jain and V. B. Semwal, "A novel feature extraction method for preimpact fall detection system using deep learning and wearable sensors," *IEEE Sensors J.*, vol. 22, no. 23, pp. 22943–22951, Oct. 2022.
[24] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "Automatic fall monitoring: A review," *Sensors*, vol. 14, no. 7, pp. 12900–12936, Jul. 2014.
[25] N. T. Newaz and E. Hanada, "The methods of fall detection: A literature review," *Sensors*, vol. 23, no. 11, p. 5212, May 2023.
[26] F. Luna-Perejón, M. J. Domínguez-Morales, and A. Civit-Balcells, "Wearable fall detector using recurrent neural networks," *Sensors*, vol. 19, no. 22, p. 4885, Nov. 2019.
[27] M. Qasaimeh et al., "Benchmarking vision kernels and neural network inference accelerators on embedded platforms," *J. Syst. Archit.*, vol. 113, Feb. 2021, Art. no. 101896.
[28] V.-R. Xefteris, A. Tsanousa, G. Meditskos, S. Vrochidis, and I. Kompatsiaris, "Performance, challenges, and limitations in multimodal fall detection systems: A review," *IEEE Sensors J.*, vol. 21, no. 17, pp. 18398–18409, Sep. 2021.
[29] M. Saleh and R. L. B. Jeannès, "Elderly fall detection using wearable sensors: A low cost highly accurate algorithm," *IEEE Sensors J.*, vol. 19, no. 8, pp. 3156–3164, Apr. 2019.
[30] H. Mankodiya et al., "XAI-fall: Explainable AI for fall detection on wearable devices using sequence models and XAI techniques," *Mathematics*, vol. 10, no. 12, p. 1990, Jun. 2022.
[31] A. Choi et al., "Deep learning-based near-fall detection algorithm for fall risk monitoring system using a single inertial measurement unit," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 2385–2394, 2022.
[32] M. M. Kabir, J. Shin, and M. F. Mridha, "Secure your steps: A class-based ensemble framework for real-time fall detection using deep neural networks," *IEEE Access*, vol. 11, pp. 64097–64113, 2023.
[33] A. Danilenka et al., "Real-time AI-driven fall detection method for occupational health and safety," *Electronics*, vol. 12, no. 20, p. 4257, Oct. 2023.
[34] H. Son et al., "A machine learning approach for the classification of falls and activities of daily living in agricultural workers," *IEEE Access*, vol. 10, pp. 77418–77431, 2022.
[35] C. Vishnu, R. Datla, D. Roy, S. Babu, and C. K. Mohan, "Human fall detection in surveillance videos using fall motion vector modeling," *IEEE Sensors J.*, vol. 21, no. 15, pp. 17162–17170, Aug. 2021.
[36] R. T. Al_Hassani and D. C. Atilla, "Human activity detection using smart wearable sensing devices with feed forward neural networks and PSO," *Appl. Sci.*, vol. 13, no. 6, p. 3716, Mar. 2023.
[37] M. J. A. Nahian et al., "Towards an accelerometer-based elderly fall detection system using cross-disciplinary time series features," *IEEE Access*, vol. 9, pp. 39413–39431, 2021.
[38] A. Chelli and M. Pätzold, "A machine learning approach for fall detection and daily living activity recognition," *IEEE Access*, vol. 7, pp. 38670–38687, 2019.
[39] M. Musci, D. De Martini, N. Blago, T. Facchinetti, and M. Piastra, "Online fall detection using recurrent neural networks on smart wearable devices," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1276–1289, Jul. 2021.
[40] A. Shahzad and K. Kim, "FallDroid: An automated smart-phone-based fall detection system using multiple kernel learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 35–44, Jan. 2019.
[41] D. Ajerla, S. Mahfuz, and F. Zulkernine, "A real-time patient monitoring framework for fall detection," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–13, Sep. 2019, doi: 10.1155/2019/9507938.
[42] R. H. Venkatnarayan, G. Page, and M. Shahzad, "Multi-user gesture recognition using WiFi," in *Proc. 16th Annu. Int. Conf. Mobile Syst., Appl., Services*, New York, NY, USA, 2018, pp. 401–413, doi: 10.1145/3210240.3210335.
[43] Y. Zheng et al., "Zero-effort cross-domain gesture recognition with Wi-Fi," in *Proc. 17th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2019, pp. 313–325.
[44] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3D tracking via body radio reflections," in *Proc. 11th USENIX Conf. Netw. Syst. Design Implement.*, Apr. 2014, pp. 317–329.
[45] F. Adib, C.-Y. Hsu, H. Mao, D. Katabi, and F. Durand, "Capturing the human figure through a wall," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–13, 2015.
[46] H. Abdelnasser, M. Youssef, and K. A. Harras, "WiGest: A ubiquitous WiFi-based gesture recognition system," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2015, pp. 1472–1480.
[47] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Device-free human activity recognition using commercial WiFi devices," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1118–1131, May 2017.
[48] (2023). *STMicroelectronics*. [Online]. Available: https://www.st.com/en/development-tools/stm32cubemx.html
[49] A. Sucerquia, J. López, and J. Vargas-Bonilla, "SisFall: A fall and movement dataset," *Sensors*, vol. 17, no. 12, p. 198, Jan. 2017.
[50] P. Goyal and M. S. Maurer, "Syncope in older adults," *J. Geriatric Cardiol. (JGC)*, vol. 13, no. 5, p. 380, 2016.
[51] M. Kangas, I. Vikman, L. Nyberg, R. Korpelainen, J. Lindblom, and T. Jämsä, "Comparison of real-life accidental falls in older people with experimental falls in middle-aged test subjects," *Gait Posture*, vol. 35, pp. 500–505, Mar. 2012.
[52] S. Hymel et al., "Edge impulse: An MLOps platform for tiny machine learning," 2022, *arXiv:2212.03332*.
[53] F. Hussain, F. Hussain, M. Ehatisham-Ul-Haq, and M. A. Azam, "Activity-aware fall detection and recognition based on wearable sensors," *IEEE Sensors J.*, vol. 19, no. 12, pp. 4528–4536, Jun. 2019.
[54] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, "A high reliability wearable device for elderly fall detection," *IEEE Sensors J.*, vol. 15, no. 8, pp. 4544–4553, Aug. 2015.

Sara Campanella (Student Member, IEEE) received the master's (cum laude) degree in biomedical engineering from the Università Politecnica delle Marche, Ancona, Italy, in July 2022, where she is currently pursuing the Ph.D. degree in biomedical, electronics, telecommunications, and nanotechnology engineering.

In November 2022, she won the "O. Carlini" scholarship, funded by Consortium GARR. Her research interests are AI approaches to optimize wearable devices, embedded AI systems, and micro- and nanoelectronic devices.



Alaa Alnasef received the bachelor's degree from Damascus University, Damascus, Syria, in 2018, and the master's degree in biomedical engineering from the Università Politecnica delle Marche, Ancona, Italy, in 2023.

He is currently a Firmware Developer for a company in Loreto, Italy. His research interests are the application of deep learning and machine learning in edge devices, sensor fusion, and embedded systems.

Laura Falaschetti (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electronic engineering from the Università Politecnica delle Marche, Ancona, Italy, in 2008, 2012, and 2016, respectively.

In 2017, she joined the Department of Information Engineering (DII), Università Politecnica delle Marche, as a Postdoctoral Research Fellow. She is currently a Postdoctoral Researcher at DII and an Assistant Professor of electronic systems with the Università Politecnica delle Marche. Her current research interests include embedded systems, machine learning, neural networks, pattern recognition, system identification, signal processing, image processing, speech processing, and biosignal analysis.

Paola Pierleoni received the master's degree in electronic engineering and the Ph.D. degree in electrical engineering from the Università Politecnica delle Marche, Ancona, Italy, in 1991 and 1995, respectively.

In 1991, she joined the Department of Information Engineering, Università Politecnica delle Marche, where she is currently an Assistant Professor of Telecommunications. Her research topics include network protocols, wireless sensor networks, the Internet of Things, signal processing, and embedded devices development.

Alberto Belli received the master's degree in telecommunications engineering and the Ph.D. degree in biomedical, electronic, and telecommunications engineering from the Università Politecnica delle Marche, Ancona, Italy, in 2012 and 2016, respectively.

He is a Technician with the Department of Information Engineering, Università Politecnica delle Marche. His research interests are wireless sensor networks for the Internet of Things, data fusion algorithms for array sensors, and wearable sensors.

Lorenzo Palma (Member, IEEE) received the master's (cum laude) degree in electronic engineering and the Ph.D. degree in information engineering from the Università Politecnica delle Marche, Ancona, Italy, in 2012 and 2017, respectively.

He is currently an Assistant Professor (RTD-B) at the Università Politecnica delle Marche. His main research interests are electronic devices, embedded artificial intelligence, wireless sensors networks, embedded systems, and the Internet of Medical Things.

Open Access funding provided by 'Università Politecnica delle Marche' within the CRUI CARE Agreement