

## Article

# Design of a Wearable Healthcare Emergency Detection Device for Elder Persons

Flora Amato, Walter Balzano and Giovanni Cozzolino \* 

Department of Electrical Engineering and Information Technology, University of Naples Federico II, 80125 Napoli, Italy; flora.amato@unina.it (F.A.); walter.balzano@unina.it (W.B.)

\* Correspondence: giovanni.cozzolino@unina.it

**Abstract:** Improving quality of life in geriatric patients is related to constant physical activity and fall prevention. In this paper, we propose a wearable system that takes advantage of sensors embedded in a smart device to collect data for movement identification (running, walking, falling and daily activities) of an elderly user in real-time. To provide high efficiency in fall detection, the sensor's readings are analysed using a neural network. If a fall is detected, an alert is sent through a smartphone connected via Bluetooth. We conducted an experimental session using an Arduino Nano 33 BLE Sense board in inside and outside environments. The results of the experiment have shown that the system is extremely portable and provides high success rates in fall detection in terms of accuracy and loss.

**Keywords:** neural networks; e-health; Arduino; wearable; machine learning



**Citation:** Amato, F.; Balzano, W.; Cozzolino, G. Design of a Wearable Healthcare Emergency Detection Device for Elder Persons. *Appl. Sci.* **2022**, *12*, 2345. <https://doi.org/10.3390/app12052345>

Academic Editors: Stefano Silvestri and Francesco Gargiulo

Received: 20 November 2021

Accepted: 18 February 2022

Published: 23 February 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Among the leading causes of severe harm and death in the elderly, there is the problem of falls at home. Lack of balance, sudden lurching and confusion after getting out of bed cause much mortality and bedriddenness. The effect is greatest in patients with dementia, who often receive harm that limits their mobility, forcing them to spend the rest of their lives bedridden or in assisted living residences.

According to the best statistics available, in Italy, people over 65 years of age fall at least once during the year. Of these, 43% fall more than once, and of these falls, over 60% occur at home. A large portion of those who fall are seniors with dementia. The bedroom accounts for as many as 25% of total falls. In the United States, falls are the leading cause of unintentional death and the 7th leading cause of death in persons aged  $\geq 65$  years. In 2018, there were 32,522 fall-related deaths of people  $\geq 65$  and only 4933 fall-related deaths of people younger than 65; thus, 85% of fall-related deaths occur in the 13% of the population who are  $\geq 65$  [1].

How a person falls will dictate the types of injuries that may result. For example, falling forward or backwards, striking the hand first as an unconditional reflex, usually causes a wrist fracture. Instead, a rupture of the hip is characteristic of falls to one side or the other. When an older adult suddenly gets out of bed, their or her body needs time to restore balance and cope with the new situation.

The main problem is not the fall and the fracture, but its consequences. In fact, in the elderly, pathologies such as osteoporosis and other physiological changes related to ageing cause slower healing, additional discomfort and effects from the psychological point of view. Many elderly patients are reluctant to report falling because they view falling as part of the ageing process or fear being restricted in their activities or hospitalised.

Falls can impair the independence of older adults and cause a range of personal and socioeconomic consequences. In fact, falls were responsible for more than 3 million emergency department visits by older persons. Medical expenditures for nonfatal fall injuries were approximately \$50 billion in 2019 and are sure to increase [2]. However,

clinicians often underestimate the damage from a fall unless the patient has an obvious injury, as history and objective examination usually do not include detailed assessments.

Anyone who lives with, helps or works with seniors, particularly those with illnesses, knows how difficult it is to get them to listen to and follow the suggestions and directions they are given. Therefore, we designed a wearable emergency recognition device for elder persons with the aim of detecting dangerous events, such as falls, in order to trigger assistance. In our case study, we used patients with metabolic disorders, who are more likely to be prone to falls.

The paper is organised as follows. Section 2 reports a discussion of related works on the fall detection topic. In Section 3, we describe our fall detection system and its constituent components. In Section 4, the experimental results and evaluation are presented. Lastly, conclusions and future work are described in Section 5.

## 2. Related Works

At present, several solutions have been proposed for elderly fall detection. Such solutions are categorised into three main types according to the sensor-technology used: non-wearable systems (NWS), wearable systems (WS) and fusion or hybrid systems (FS).

In particular, NWS systems [3–5] use vision-based sensors strategically distributed in the home of the elder. They have been proven powerful and robust at detecting falls; however, these systems have high costs, can be obviously be effective only in indoor environments and could generate privacy issues for the elders or the people that assist them.

To overcome these limitations, WS systems were proposed. They typically use inertial sensors, such as an accelerometer or gyroscope, usually attached to the elder for motion detection. Accelerometers are being increasingly used in WS systems because they offer advantages: low power consumption; affordability; lightness; ease of use; small size; the potential to be mounted on various body parts; and most importantly, extreme portability. Therefore, in some representative papers [6–8] a 3-axis accelerometer with the threshold-based algorithm was used. In these papers, the authors detected falls when the acceleration from a 3-axis accelerometer exceeded the threshold. One of the essential advantages of using the threshold-based method is that it is less complex and less computationally intensive than the other methods. However, finding suitable thresholds to detect all types of falls without mislabelling activities of daily living (ADL) has proven to be a complex problem.

A similar approach uses a smartphone's built-in accelerometer to monitor the movement data of an elderly person continuously. In [9], the collected data were used to test three different learning classifiers offline: decision trees, k-nearest-neighbours (KNN) and naive Bayes. The results show that the decision-trees-based algorithm had the best performance, with more equilibrated sensitivity and specificity values compared with the other algorithms. Nevertheless, due to smartphones' relatively high energy consumption, this system could only be active for a short period.

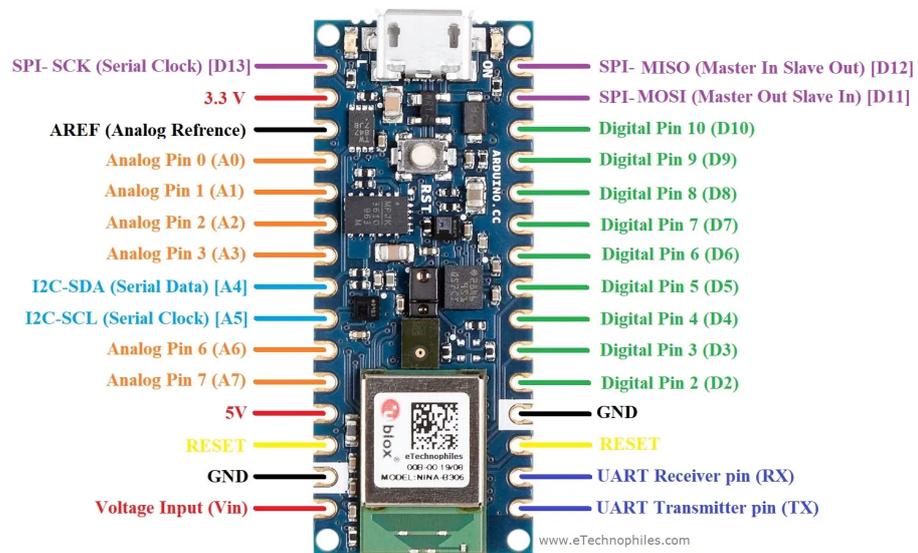
Recently, WS systems based on machine learning (ML) approaches have been proposed [10–14] to address these limitations and improve the accuracy of fall detection. One study [15] used a nonlinear support vector machine to extract features and gain meaning from body data captured by an accelerometer attached to a smart textile. Two feature extractions were required to identify the peak to detect the fall direction, requiring more processing than a single extraction algorithm. The authors of [16] detected and predicted falls using a method based on the hidden Markov model (HMM), which involved gathering time series from the movements obtained by a three-axis accelerometer placed on the upper body. The test results show a perfect success rate of drop detection (100% sensitivity and 100% specificity). However, they used data samples from adolescents' simulated activities to train and adjust the HMM and the system's thresholds.

## 3. Fall Detection System

Our methodology foresees developing a wearable system for detecting falls of older people, which takes advantage of low-power smart devices' capabilities and a neural

network for movement detection recognition. In this work, we have followed an ML approach by using a neural network for fall detection, but we differ from related works in the system's design.

While other related works exploit multiple sensors that collect movements and send data to a device that analyse them, in this work, we used a single device for activity monitoring and recognition through a neural network deployed on an Arduino nano 33 BLE Sense board. The board has a small size of  $45 \times 18$  mm, which makes it suitable for prototype wearables, and is equipped with several integrated sensors to measure environmental variables. In Figure 1, one can see the board's main components and input/output interfaces. This choice brings versatility and portability advantages, since the other related works' solutions are constrained to indoor environments that rely on non-portable infrastructures or require multiple sensors to be worn on the body.



**Figure 1.** Arduino 33 Nano BLE Sense pinout.

Another innovation of our work is that the monitoring board interacts with a smartphone to collect and manage events. The board can communicate with the smartphone through a Bluetooth Low Energy (BLE) module: when it detects a fall, it sends a notification to the smartphone. To avoid false alarms, a mobile application on the smartphone manages the notifications, asking to the user if there is an emergency. If no response is provided within 60 s, the smartphone forwards an alert by calling a healthcare professional and sending information about the location of the older person. Furthermore, the detected events are stored on the smartphone—one the one hand to give more accurate information to healthcare professionals and on the other hand to provide an efficient way to enhance the neural network's training, together with feedback provided in response to detected events.

### 3.1. Datasets

The analysis of the recent related literature showed that current studies tend to prefer the use of already existing public repositories containing falls and ADLs, although no particular dataset can be considered a globally accepted benchmarking tool. For neural network training, we used two different datasets.

The first dataset chosen for recognition of falls and daily activities (ADL) [17] includes 11 activities and three trials for each of them. For the dataset collection, 17 different subjects performed six different activities of daily living (walking, standing, lifting an object, sitting and lying down) and five different types of falls (falling forward using hands, falling forward using knees, falling backwards, falling sitting, and falling sideways). Data were collected with a multi-modal approach using wearable sensors, ambient sensors and vision

devices. For data consistency, we selected a single subject and chose to refer only to the data acquired by inertial sensors placed on the right wrist. For each type of activity (excluding walking), we selected five samples for a total of 50 samples for falls and daily activities, respectively.

We used Power BI to eliminate redundant data and measurements unrelated to the right wrist or related to the three axes of acceleration and gyroscope. Therefore, the resulting dataset was trimmed to be used in the training of the neural network that we show in detail in the dedicated section.

The *Run or Walk* dataset [18] contains running and walking data collected from iOS devices. Initially, the dataset consisted of a single file representing 88,588 data samples collected by the device's accelerometer and gyroscope during an interval of 10 s and at a frequency of approximately 5.4/s. For each row, there is an activity type represented by "activity" column which acts as label and a "wrist" column which represents the wrist whereupon the device was placed to collect samples. Specifically, each row of the dataset contained:

- acceleration\_x;
- acceleration\_y;
- acceleration\_z;
- gyro\_x;
- gyro\_y;
- gyro\_z;
- label "0" for walking;
- label "1" for running;
- label "0" for the left wrist;
- label "1" for the right wrist.

The original dataset also contained the columns "date", "time" and "username", which for obvious reasons, have been eliminated by PowerBI. Moreover, we chose to consider the measurements made only on the right wrist (for consistency with data collected for the others activities). Then, we collected 50 samples for walking and 50 for running. Angular velocity values were transformed from rad/s to deg/s to align them with the fall/adl dataset values. Therefore, all values contained in the gyro axes columns were multiplied by  $57.2958^\circ/\text{rad}$ .

### 3.2. Data Pre-Processing

When data are transmitted to the designed neural network, the size of each input datum should be the same as the number of input layer variables (in our model, the input layer size is 300). However, since the duration of each action, including falling, is different, we needed to make the sizes of the data the same. Based on this, all data were unified to have 50 values.

We split complete dataset into four lists of gestures, namely, "adl", "fall", "walk" and "run"; we had 50 samples for each gesture. For each of these files we had to normalise input data between 0 and 1 in order to create tensors. Each row contained normalised input data, coming from acceleration and angular velocity values, and the output represented by an eye matrix encoding the expected activity value. Data pre-processing reported in the following Listing 1.

**Listing 1.** Dataset parsing and pre-processing.

```

1 # Set a fixed random seed value, for reproducibility, this will allow us to
  # get the same random numbers each time the notebook is run
2 SEED = 1337
3 np.random.seed(SEED)
4 tf.random.set_seed(SEED)
5
6 # the list of gestures that data is available for
7 GESTURES = ["adl", "fall", "walk", "run",]
8 NUM_GESTURES = len(GESTURES)
9 SAMPLES_PER_GESTURE = 50
10
11 # create a one-hot encoded matrix that is used in the output
12 ONE_HOT_ENCODED_GESTURES = np.eye(NUM_GESTURES)
13
14 inputs = []
15 outputs = []
16
17 # read each csv file and push an input and output
18 for gesture_index in range(NUM_GESTURES):
19     gesture = GESTURES[gesture_index]
20     print(f"Processing index {gesture_index} for gesture '{gesture}'.")
21
22     output = ONE_HOT_ENCODED_GESTURES[gesture_index]
23
24     df = pd.read_csv("/content/" + gesture + ".csv")
25
26     # calculate the number of gesture recordings in the file
27     num_recordings = int(df.shape[0] / SAMPLES_PER_GESTURE)
28
29     print(f"\tThere are {num_recordings} recordings of the {gesture} gesture.")
30
31     for i in range(num_recordings):
32         tensor = []
33         for j in range(SAMPLES_PER_GESTURE):
34             index = i * SAMPLES_PER_GESTURE + j
35             # normalize the input data, between 0 and 1:
36             tensor += [
37                 (df['aX'][index] + 4) / 8,
38                 (df['aY'][index] + 4) / 8,
39                 (df['aZ'][index] + 4) / 8,
40                 (df['gX'][index] + 1000) / 2000,
41                 (df['gY'][index] + 1000) / 2000,
42                 (df['gZ'][index] + 1000) / 2000
43             ]
44
45         inputs.append(tensor)
46         outputs.append(output)
47
48 # convert the list to numpy array
49 inputs = np.array(inputs)
50 outputs = np.array(outputs)
51
52 print("Data set parsing and preparation complete.")

```

**3.3. Training, Testing and Validation Datasets**

For model training we randomly split input and output pairs into a training set (60%), a testing set (20%) and a validation set (20%), as reported in the following Listing 2.

**Listing 2.** Dataset randomisation and splitting.

```

1 # Randomise the order of the inputs, so they can be evenly distributed for
   training, testing, and validation
2 num_inputs = len(inputs)
3 randomise = np.arange(num_inputs)
4 np.random.shuffle(randomise)
5
6 # Swap the consecutive indexes (0, 1, 2, etc) with the randomised indexes
7 inputs = inputs[randomise]
8 outputs = outputs[randomise]
9
10 # Split the recordings (group of samples) into three sets: training, testing
   and validation
11 TRAIN_SPLIT = int(0.6 * num_inputs)
12 TEST_SPLIT = int(0.2 * num_inputs + TRAIN_SPLIT)
13
14 inputs_train, inputs_test, inputs_validate = np.split(inputs, [TRAIN_SPLIT,
   TEST_SPLIT])
15 outputs_train, outputs_test, outputs_validate = np.split(outputs, [
   TRAIN_SPLIT, TEST_SPLIT])
16
17 print("Data set randomisation and splitting complete.")

```

### 3.4. Model Training

In this work, we used Google TensorFlow for neural network configuration and learning, since it makes neural network implementation convenient, because it provides functions used for machine learning, including activation functions and an initialisation function.

To exploit the advantages of neural networks while keeping a simple model, able to be deployed on the Arduino board, we designed a sequential neural network model for ADL recognition consisting of:

- A dense layer with 50 neurons and a *sigmoid* activation function;
- A dense level with 25 neurons and a *sigmoid* activation function;
- A final level with four neurons and an activation function *softmax*.

The following Listing 3 shows the implemented model:

**Listing 3.** Neural network model definition and training.

```

1 # build the model and train it
2 model = tf.keras.Sequential()
3 model.add(tf.keras.layers.Dense(50, activation='ReLU'))
4 model.add(tf.keras.layers.Dense(25, activation='ReLU'))
5 # softmax is used, because we only expect one gesture to occur per input
6 model.add(tf.keras.layers.Dense(NUM_GESTURES, activation='softmax'))
7 model.compile(optimizer='adam', loss='mse', metrics=['accuracy'])
8 history = model.fit(inputs_train, outputs_train, epochs=80, batch_size=1,
   validation_data=(inputs_validate, outputs_validate))

```

Since the sample size of the experiment was 50, and we had three components for acceleration and angular velocity per sample ( $x$ ,  $y$  and  $z$ ;  $gx$ ,  $gy$  and  $gz$ ), the number of variables in the input layer was set to  $50 \times 6$ . In the hidden layers, a ReLU function was used as the activation function, for performance reasons.

The output layer consists of four neurons—[0, 1]—for the “adl”, “fall”, “walk” and “run” activities. In the output stage, the activation function is *softmax*, so the sum of the output probabilities has to be 1. In our case, having four different classes, we obtained a probability for each of them. The predicted movement is the one with the highest probability.

### 3.5. Model Deployment

The model was built and trained using the TensorFlow and Keras libraries. The obtained model was converted to a Tensor Flow Lite version, as reported in Listing 4, suitable to be loaded into the Arduino IDE and then flashed into the board. Thus we built

a classifier that prints a prediction on a serial monitor and sends emergency notifications through Bluetooth messages to a smartphone.

**Listing 4.** Neural Network conversion into Tensorflow Lite model.

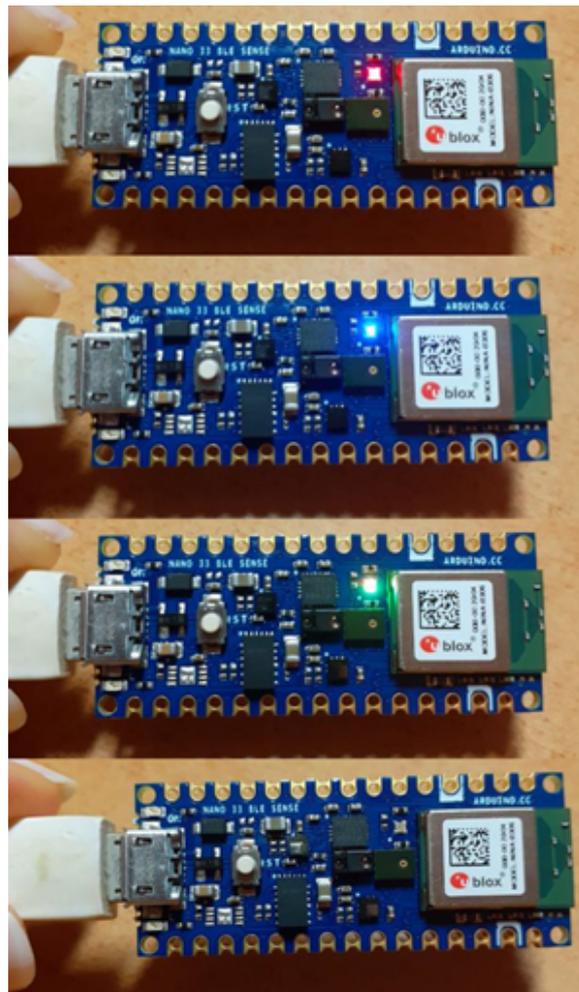
```

1 # Convert the model to the TensorFlow Lite format without quantization
2 converter = tf.lite.TFLiteConverter.from_keras_model(model)
3 tflite_model = converter.convert()
4
5 # Save the model to disk
6 open("model.tflite", "wb").write(tflite_model)
7
8 import os
9 basic_model_size = os.path.getsize("model.tflite")
10 print("Model is %d bytes" % basic_model_size)

```

The classifier implemented for the Arduino board predicts four possible motions (as illustrated above). The detection of a movement is signalled by turning on the RGB LED of the board, as shown in Figure 2, according to the following scheme:

- Red LED, when a fall is detected;
- Blue LED, for running;
- Green LED, for walking;
- LED off, for actions of daily life (ADL).



**Figure 2.** From the top: falling, running, walking and standing detection.

#### 4. Results

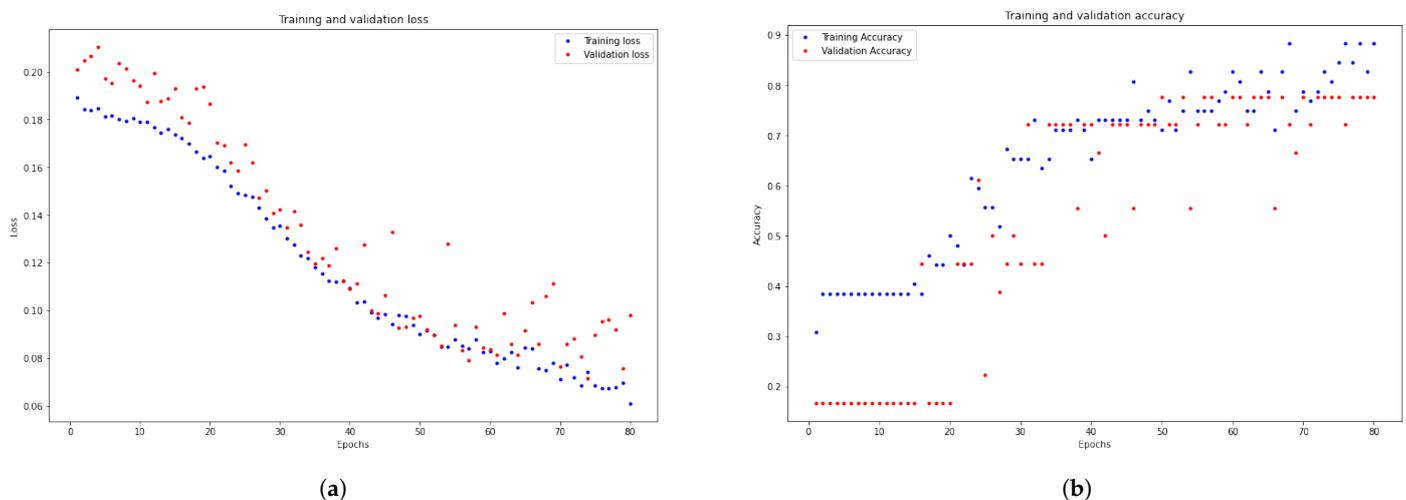
How to evaluate fall detection systems (FDSs) in realistic conditions is still an unresolved experimental problem. The main users of FDSs are supposed to be the elderly. The current public databases containing actual falls experienced by older adults are certainly scarce. In our scenario of monitoring older persons' activities (generally people with limited mobility), falls could be identified as movements that clearly deviate from the detected patterns of the samples in the training set. Anyway, in the absence of measurement repositories with a significant number of actual falls, experiments were conducted to obtain the acceleration values of falls. We excluded older people from falls simulations because they could have resulted in severe injuries to such subjects.

For fall simulations, we prepared an experimental environment consisting of a floor mat capable of absorbing one's fall on which we put an unstable platform. The subject, to which the smart device was attached on the right wrist by a strip string, was asked to stand on the unstable platform. By slightly moving the platform, the subject's fall was induced.

The model for motion detection (ADL, fall, walk, run) was trained for 80 epochs, obtaining the results shown in Figure 3.

The parameters shown are:

- *loss*, defined as the root mean square error between the actual value and the predicted value during training;
- *accuracy*, as the percentage of correct predictions, compared to the total predictions during training;
- *val\_loss*, loss on the validation data;
- *val\_accuracy*, accuracy on the validation data.



**Figure 3.** Training of the movement detection model: (a) *loss* and *val\_loss* functions for the movement classifier. (b) *accuracy* and *val\_accuracy* functions for the movement classifier.

Figure 3a shows the loss and *val\_loss* obtained for the motion classifier. On the validation data, the loss function was 0.10. Figure 3b shows the accuracy and *val\_accuracy* obtained for the motion classifier. On the validation data, the accuracy did not get beyond 78%.

For a real-world validation session, we tested an elderly person performing ADLs and walking, since they are safe experiments. These data are more representative of the posture, walking speed and other factors typical of older people.

The subject performed ADLs wearing the device for a week and provided feedback on the alert notifications prompted by the smartphone. The subject also performed a prefixed set of activities at the end of each day to compare the results. We estimated network classification performance at the beginning and the end of the experiment by evaluating samples of data collected during test set of activities performed on day 1 and day 7. We reported the results in Figure 4.

Figure 4a shows the confusion matrix obtained by the model during day 1. The recognition of fall, walk and run activities were good (all five running activities were recognised, along with the four walking activities and the two falling activities). However, in the case of the ADLs, due to their variability, only one was recognised adequately, and the remainder were predicted to be falls. Figure 4b shows the model’s confusion matrix obtained during day 7, after tuning the network with feedback provided by the subject through the smartphone. Fall, walk and run recognition were still good; and in ADLs recognition, 50% of events were correctly recognised.

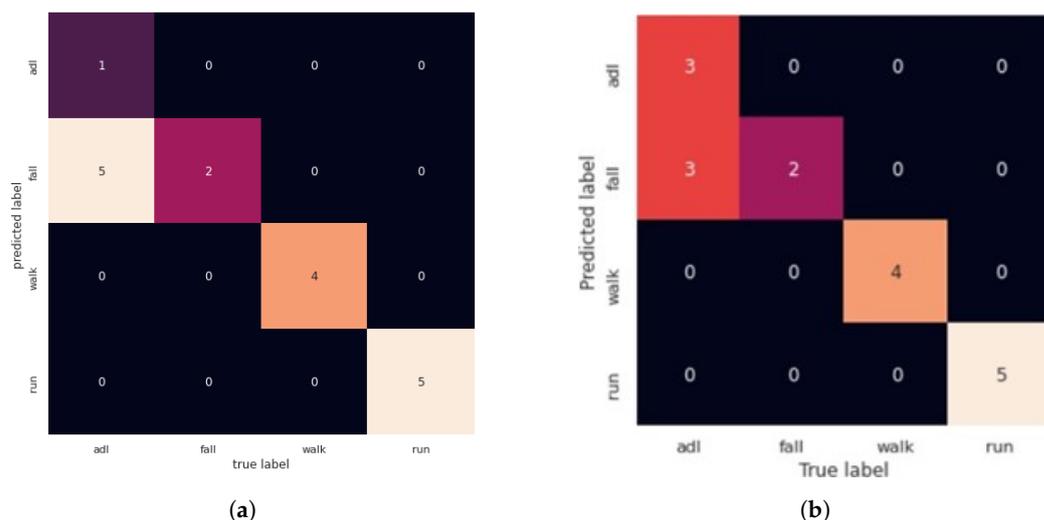


Figure 4. Model’s confusion matrix. (a) Confusion matrix for the movement detection classifier on 1. (b) Confusion matrix for the movement detection classifier on day 7.

### 5. Conclusions

In this work we have presented a system for fall detection for elderly people. The system exploits a smart sensor board on which we use a neural network trained to recognise and monitor the activity of the patient. The board interacts with a smartphone application, connected through Bluetooth with the board, which is responsible for getting user feedback to supposed fall events and forwarding emergency calls if necessary.

In previous fall detection studies, falls have recognised using acceleration sensors on the waist or the chest, and the recognition rate has been over 95%. However, when an acceleration sensor on the wrist was used, the recognition rate was about 75%. The artificial neural network proposed in this work was able to recognise activities with 78% accuracy using the acceleration of the wrist. This is a relatively small improvement compared to the conventional fall detection mechanism, which is due to the simple neural network model that was designed to suit the limited computational capabilities of such devices.

However, with wrist-band type devices, we can cut down the system costs (we may use existing smart-watches or bands) and provide comfort to the user. Moreover the proposed system is portable, usable in outdoor environments and upgradeable through the firmware. Furthermore, the system analyses sensor data with an embedded computational unit (CU), not having the need for streaming data to an external CU, thereby preventing draining of the battery of the connected smartphone. The latter is only responsible for obtaining feedback from the user and forwarding emergency notifications.

In future developments we will provide the ability to classify more activities, so that the living patterns of older persons can be better recognised. Furthermore, we could also integrate speech recognition features to recognise help requests, including those not related to falls [19,20]. Moreover, we foresee the need to apply security and privacy techniques in order to process the data acquired by the sensors [21,22].

**Author Contributions:** Conceptualization, F.A., W.B. and G.C.; methodology, F.A., W.B. and G.C.; software, F.A., W.B. and G.C.; validation, F.A., W.B. and G.C.; formal analysis, F.A., W.B. and G.C.; data curation, F.A., W.B. and G.C.; writing—original draft preparation, F.A., W.B. and G.C.; writing—review and editing, F.A., W.B. and G.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** <https://sites.google.com/up.edu.mx/har-up/>, <https://www.kaggle.com/vmalyi/run-or-walk> (accessed on 18 November 2021).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Older Adult Falls. Available online: <https://injuryfacts.nsc.org/home-and-community/safety-topics/older-adult-falls/> (accessed on 28 November 2021).
- Older Adult Fall Prevention. Available online: <https://www.cdc.gov/falls/> (accessed on 28 November 2021).
- Harrou, F.; Zerrouki, N.; Sun, Y.; Houacine, A. Vision-based fall detection system for improving safety of elderly people. *IEEE Instrum. Meas. Mag.* **2017**, *20*, 49–55. [CrossRef]
- Solbach, M.D.; Tsotsos, J.K. Vision-based fallen person detection for the elderly. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 1433–1442.
- De Miguel, K.; Brunete, A.; Hernando, M.; Gambao, E. Home camera-based fall detection system for the elderly. *Sensors* **2017**, *17*, 2864. [CrossRef] [PubMed]
- Dias, P.V.G.; Costa, E.D.M.; Tcheou, M.P.; Lovisolio, L. Fall detection monitoring system with position detection for elderly at indoor environments under supervision. In Proceedings of the 2016 8th IEEE Latin-American Conference on Communications (LATINCOM), Medellin, Colombia, 16–18 November 2016; pp. 1–6.
- Phu, P.T.; Hai, N.T.; Tam, N.T. A threshold algorithm in a fall alert system for elderly people. In Proceedings of the 5th International Conference on Biomedical Engineering in Vietnam, Ho Chi Minh City, Vietnam, 16–18 June 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 347–350.
- Santiago, J.; Cotto, E.; Jaimes, L.G.; Vergara-Laurens, I. Fall detection system for the elderly. In Proceedings of the 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 9–11 January 2017; pp. 1–4.
- Aguiar, B.; Rocha, T.; Silva, J.; Sousa, I. Accelerometer-based fall detection for smartphones. In Proceedings of the 2014 IEEE International Symposium on Medical Measurements and Applications (MeMeA), Lisbon, Portugal, 11–12 June 2014; pp. 1–6.
- Pannurat, N.; Thiemjarus, S.; Nantajeewarawat, E. Automatic fall monitoring: A review. *Sensors* **2014**, *14*, 12900–12936. [CrossRef] [PubMed]
- Hwang, J.Y.; Kang, J.; Jang, Y.W.; Kim, H.C. Development of novel algorithm and real-time monitoring ambulatory system using Bluetooth module for fall detection in the elderly. In Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, San Francisco, CA, USA, 1–5 September 2004; Volume 1, pp. 2204–2207.
- Noury, N.; Hervé, T.; Rialle, V.; Virone, G.; Mercier, E.; Morey, G.; Moro, A.; Porcheron, T. Monitoring behavior in home using a smart fall sensor and position sensors. In Proceedings of the 1st Annual International IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology. Proceedings (Cat. No. 00EX451), Lyon, France, 12–14 October 2000; pp. 607–610.
- Hussain, F.; Ehatisham-ul Haq, M.; Azam, M.A.; Khalid, A. Elderly assistance using wearable sensors by detecting fall and recognizing fall patterns. In Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers, Singapore, 8–12 October 2018; pp. 770–777.
- Gharghan, S.K.; Mohammed, S.L.; Al-Naji, A.; Abu-AlShaeer, M.J.; Jawad, H.M.; Jawad, A.M.; Chahl, J. Accurate fall detection and localization for elderly people based on neural network and energy-efficient wireless sensor network. *Energies* **2018**, *11*, 2866. [CrossRef]
- Mezghani, N.; Ouakrim, Y.; Islam, M.R.; Yared, R.; Abdulrazak, B. Context aware adaptable approach for fall detection bases on smart textile. In Proceedings of the 2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), Orlando, FL, USA, 16–19 February 2017; pp. 473–476.
- Tong, L.; Song, Q.; Ge, Y.; Liu, M. HMM-based human fall detection and prediction method using tri-axial accelerometer. *IEEE Sens. J.* **2013**, *13*, 1849–1856. [CrossRef]
- Martínez-Villaseñor, L.; Ponce, H.; Brieva, J.; Moya-Albor, E.; Núñez-Martínez, J.; Peñafort-Asturiano, C. UP-fall detection dataset: A multimodal approach. *Sensors* **2019**, *19*, 1988. [CrossRef] [PubMed]
- Run or Walk—A Dataset Containing Labeled Sensor Data from Accelerometer and Gyroscope. Available online: <https://www.kaggle.com/vmalyi/run-or-walk> (accessed on 18 November 2021).

19. Silvestri, S.; Gargiulo, F.; Ciampi, M.; De Pietro, G. Exploit multilingual language model at scale for ICD-10 clinical text classification. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–7.
20. Ciampi, M.; De Pietro, G.; Masciari, E.; Silvestri, S. Some lessons learned using health data literature for smart information retrieval. In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, 30 March–3 April 2020; pp. 931–934.
21. Amato, F.; Casola, V.; Cozzolino, G.; De Benedictis, A.; Mazzocca, N.; Moscato, F. A Security and Privacy Validation Methodology for e-Health Systems. *ACM Trans. Multimedia Comput. Commun. Appl.* **2021**, *17*, 1–22. [[CrossRef](#)]
22. Amato, F.; Castiglione, A.; Cozzolino, G.; Narducci, F. A semantic-based methodology for digital forensics analysis. *J. Parallel Distribut. Comput.* **2020**, *138*, 172–177. [[CrossRef](#)]