Remote fall detection system for elderly people using non-invasive technologies

Mihai APOSTOL, Răzvan-Alexandru DUȚESCU

National Institute for Research & Development in Informatics – ICI Bucharest, Romania

mihai.apostol@ici.ro, razvan.dutescu@ici.ro

Abstract: Fall detection systems are designed to provide rapid response and assistance in the event of a fall. This article focuses on the design and implementation of the alert system for a fall detector. The system uses sensors to detect falling and quickly sends an alert to a designated caregiver or emergency services. The alert system is designed to be reliable, efficient and user-friendly, providing peace of mind for elderly or disabled individuals who are at high risk of falls. The application can manage hundreds of different sensor types and provides a user-friendly interface for configuring and managing the fall detection system, including setting up alert recipients, defining emergency protocols, individuals, and sensors and monitoring the status of the system.

Keywords: Alert systems, Eldercare, Django, Raspberry Pi, Fall detection.

Sistem pentru detectarea de la distanță a căderilor persoanelor în vârstă, bazat pe utilizarea tehnologiilor non-invazive

Rezumat: Sistemele de detectare a căderii sunt concepute pentru a oferi răspuns și asistență rapidă în cazul unei căderi. Acest articol se concentrează pe proiectarea și implementarea sistemului de alertă pentru un detector de cădere. Sistemul folosește senzori pentru a detecta căderea și trimite rapid o alertă unui îngrijitor desemnat sau serviciilor de urgență. Sistemul de alertă este conceput pentru a fi fiabil, eficient și ușor de utilizat, oferind liniște sufletească persoanelor în vârstă sau cu dizabilități care prezintă un risc ridicat de cădere. Aplicația poate gestiona sute de senzori diferiți și oferă o interfață ușor de utilizat pentru configurarea și gestionarea sistemului de detectare a căderii, inclusiv configurarea destinatarilor de alertă, definirea protocoalelor de urgență, a persoanelor și a senzorilor și monitorizarea stării sistemului.

Cuvinte cheie: sistem de alertă, django, îngrijirea bătrânilor, Raspberry Pi, detecția căderii.

1. Introduction

According to World Health Organization, the world population over 60 years old will increase from 1 billion in 2020 to 1.4 billion in 2030 (World Health Organization, 2022). This significant growth induces the need for reliable monitoring systems that require minimal supervision as most of the elder population ends up living alone and faces multiple difficulties such as mobility, eating and other everyday activities. Moreover, the elderly are prone to accidents such as falling which is a significant challenge in their day-to-day life.

A typical fall detection system (Figure 1) consists of several IoT devices such as data acquisition sensors (temperature, pressure, radar etc.), a microcontroller (such as Raspberry Pi, Arduino etc.) and an alert system. With the data acquired from the real-world environment, the microcontroller can then process the signals coming in and the alert system has the task of sending alert messages or emails to a responsible individual. A smartphone or a smartwatch can count as a complete fall detection system as these devices are usually equipped with sensors and can execute applications taking the roles of the microcontroller and the alert system.

https://doi.org/10.33436/v33i1y202303

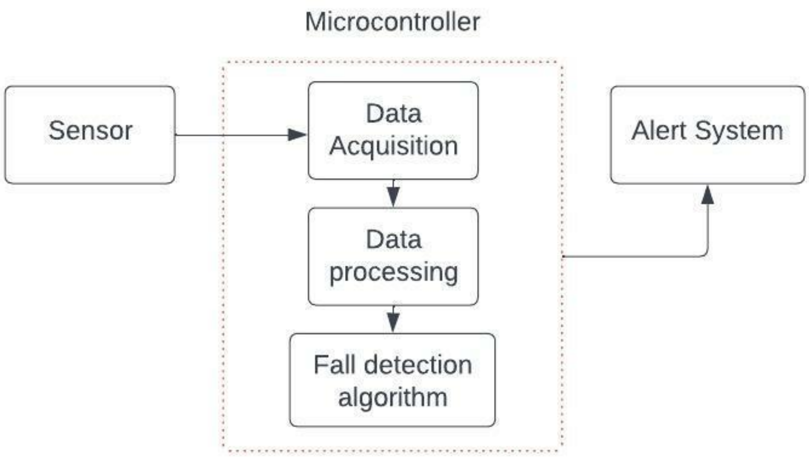


Figure 1. A typical fall detection system

There are two types of fall detection systems:

- A single sensor-based system that has only one sensor or module for data acquisition. The collected data is then processed by a fall detection algorithm which can be either a threshold-based method, a statistical model or a machine learning model;
- Multiple sensor-based systems rely on multiple sensors that capture environment data and uses threshold-based fall detection algorithms or machine learning models.

IoT devices are easy to work with and have a wide range of interoperability with similar devices. Most IoT manufacturers provide programming support by releasing specific ready to use libraries and modules usually for Python and C++. All these factors work in tandem to create system modularity, allowing developers to easily update their system with new hardware or to new requirements.

In this paper we will present the design, the technologies and all the steps of such a monitoring system that is easy to scale up or down and takes full advantage of the interoperability of IoT sensors in order to easily fit the requirements of each user. Moreover, while our proposed method does not require huge datasets for training due to not having an underlaying machine learning model, thanks to its modularity it can be easily replaced with such an algorithm.

2. Related work

Fall detection methods, systems and algorithms have been researched and developed for some time and they have been achieved in multiple ways with an emphasis on the device that detects the motion being analysed. The current state of fall detection hardware is mainly based on wearable devices such as accelerometers, magnetometers, gyroscopes, inertial measuring units or surface electromyography (Jiang et al., 2017) that have the advantage of collecting data outside the livable environment and are usually embedded in smartphones that do not require specialised equipment. Non-wearable devices which include cameras, infrared sensors, laser range scanners, pressure sensors and ground reaction force sensors have limited area coverage and some of them (such as video surveillance) are generating privacy concerns. This is the main reason that non-wearable devices are used less frequently for fall detection (Chaudhuri et al., 2014).

The most common detection techniques are threshold-based methods and machine-learning methods (Igual et al., 2013). Threshold methods usually have a lower detection rate but have a faster implementation time and do not require a lot of computer resources as opposed to machine learning algorithms. On the other hand, machine learning methods require large data sets. Furthermore, simulated datasets used for training were performed by subjects under 40 years old which is not an accurate representation of a real-life scenario with an older person. The few authentic sets that exist contain an insignificant number of entries (Usmani et al., 2021).

There are studies available that analyse dozens of fall detection systems and algorithms with a declared performance between 70% and 100% in successfully detecting a fall (Igual et al., 2013). These studies included sensors like cameras, microphones, accelerometers, radios, pressure sensors or infrared sensors. Most smartphone-based fall detector performances range from 73% to 81% while smartwatch devices can have a very high performance, most of them can perform around 95%-100% with their main drawback being that they generate privacy violations.

Alert systems are usually integrated into the fall detection system's microcontroller as a script which sends messages in the form of an email message or SMS. However, there are numerous available fall detection systems based on smartphones and smartwatches where the alert system is embedded into the device.

3. Methodology of system operation

At the macro level, the fall detection system described in Figure 2 is radio sensor-based and uses a thresholding algorithm, a Raspberry Pi microcontroller and a web application responsible for managing and sending alerts.

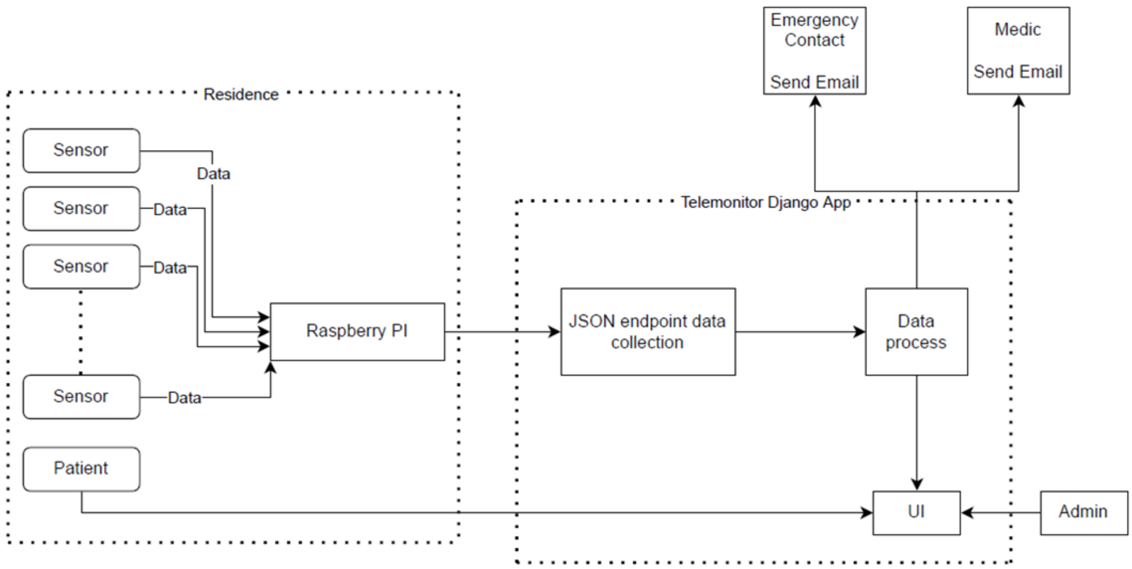


Figure 2. Our proposed fall detection system

The sensor is a Walabot Creator three-dimensional radio-frequency sensor that uses an array of 15 sensors to capture data. We used a Raspberry Pi model 4 single-board computer as the microcontroller which receives raw data, applies the fall detection algorithm and sends alerts to the alert system web application endpoint as JSON data. These alerts each have a specific code, ranging from 0 to 4 as detailed in Table 1.

Introducing a different sensor into the system only requires updating the fall detection algorithm and the alert transmission to the alert system (the JSON data). This opens up new possibilities for the fall detection system as elderly people can faint due to extreme environmental variables such as temperature, pressure, humidity or combinations of any of them.

Table 1. Alert codes representation

Room sensor type	Alert code 0 (state 0)	Alert code 1 (state 1)	Alert code 2 (state 2)	Alert code 3 (state 3)	Alert code 4 (state 4)
Proximity	No alert	Entered the room	Sat down	Fall	Exited the room
Temperature	No alert	Low temperature	High temperature	N/A	N/A

Humidity	No alert	Low humidity	High humidity	N/A	N/A
Pressure	No alert	Low pressure	High pressure	N/A	N/A
Sound	No alert	Fall	N/A	N/A	N/A

The web application was developed using the Python programming language and Django framework (Django, 2022). It is capable of sending alerts as emails or SMS and it can manage multiple and different sensor types, subjects, and rooms. All supported sensors in Table 1 are using non-invasive technology as they are not wearable.

The Walabot sensor responsible for detection is strategically placed in the room to cover its entire surface and to monitor the activities of the occupant of the room. The signal transmitted by the sensor is then processed by the Raspberry Pi. Based on the returned data, it will determine what occurred in the room and what course of action to take moving forward. One important detail that must be taken into account is the placement of the sensor. Walabot sensor calibration requires exact measurements of room dimensions (length, width and height). We must consider that there is a big difference between the sensor being placed on the ceiling or the wall as the fall detection algorithm is influenced by such details.

The transition from one state to another is determined by a Finite Automaton which is detailed in Figure 3. The states can be defined in 3 categories. The first category, the neutral state, is represented by Code 0. In this state, the sensor is at rest and the recordings made by it are done at a longer time interval. The moment a person enters the room, the sensor detects movement and switches from the neutral state to the active state. In this state, more calculations are done in the background. First, the sampling rate of the sensor is higher than the neutral state. Second, when the object appears in frame, the coordinates at which it entered the room are stored in memory, in order to compare them later when the object exits the frame. Active state includes several codes: code 1, code 2 and code 4. These are the activities that a person can do in a room. It is important to differentiate between a person who is simply sitting in a chair or lying on a bed and a person who has fallen. To make this difference we set two thresholds at different heights. If the object we are tracking is above these thresholds, we can assume that the person is standing. If said tracked object is between the two thresholds, then the person is sitting on the chair, and if the object is under both, it means that it must be checked if the person has fallen. If it has been identified as a dropped object, then the system enters the alert state and forwards code 3, sending alerts to the system. To go from the active state to the neutral state, the tracked object must exit the frame at the same point from which it entered, or in its proximity. To determine how close it is, the Euclidean distance between the entry point and the last recorded coordinates is calculated.

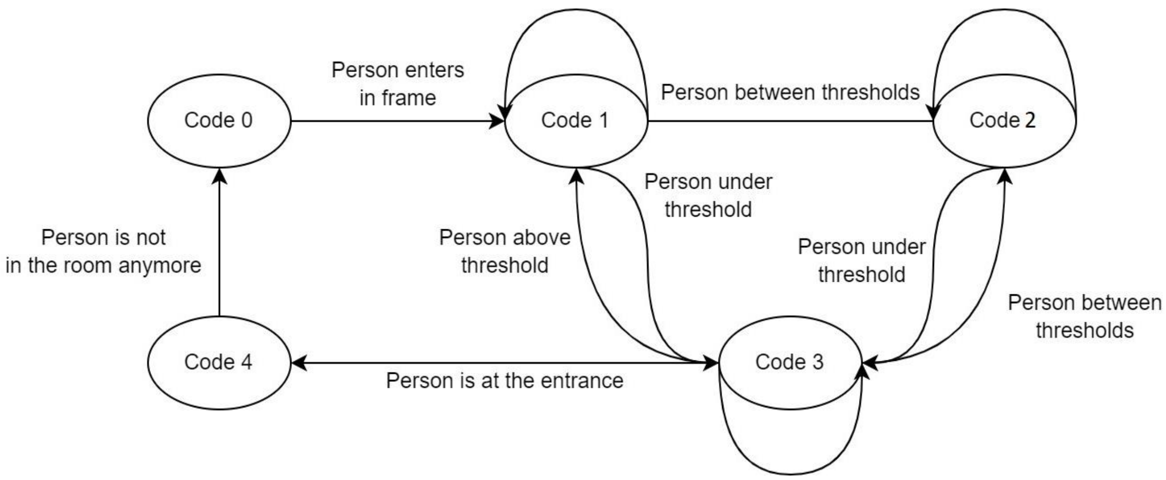


Figure 3. Finite Automaton describing the system behavior

4. Alert system

4.1. Technologies used

- Python - is a high-level general-purpose interpreted programming language. It supports multiple programming paradigms like object-oriented programming, functional programming and procedural programming. It has a wide range of applications including web development, scientific computing, data analysis, artificial intelligence and more. Python can be used as a scripting language or to build standalone applications. We used this programming language for several tasks: the base of our alert system is written in Python, the fall detection algorithm is implemented using this programming language and also sending the data is accomplished using a python script.
- Django framework - is a high-level, open-source web framework for Python. It was developed to make it easier for web developers to build secure, scalable and maintainable web applications. Django follows the model-template-view (MTV) architectural pattern and provides a full-stack framework, including an ORM for handling database interactions. This backend framework was chosen for its easy modularity, database model creation, data parsing, validations, message sending and all the logic that assembles an alert system.
- HTML (Hyper Text Markup Language) - is the standard markup language for creating web pages. It provides the structure and content of web pages and is used in conjunction with CSS and JavaScript to create dynamic and interactive websites. It is an essential technology for building websites and is widely used by web developers and designers worldwide (MDN Web Docs, 2022b). No websites or web application can exist without HTML Moreover, there is no alternative to this technology, only frameworks that are built upon HTML.
- CSS (Cascading Style Sheets) - is a stylesheet language used for describing the look and formatting of a document written in HTML. It allows web developers to separate the presentation of a website from its content, making it easier to maintain and update the visual design (MDN Web Docs, 2022a). This is the main technology used for creating the user interface with minimum interactivity.
- JavaScript - is a high-level, interpreted programming language that is primarily used to create dynamic, interactive and responsive web pages. It is a client-side language, meaning that the code is executed by the user's browser rather than on the server. JavaScript is commonly used for adding interactivity to websites, such as form validation, creating drop-down menus, and creating animations (MDN Web Docs, 2022c). JavaScript technology was used for implementing front-end validations, confirmations and enhance user experience.

4.2. Alert system architecture

The alert system is capable of handling hundreds of sensors, hence, it can look after a large number of old people since the alert system is built in such a way that it can be scaled up to suit those needs. Alerted entities can be any designated person (relatives, friends etc.) or a supervising medic depending on the subject’s medical health problems. The alert can arrive via an email message or SMS.

Figure 4 reflects the overall database model. A registered user of this platform can be either an administrator responsible for the application maintenance (user registration, adding sensors etc.) or it can be the patient.

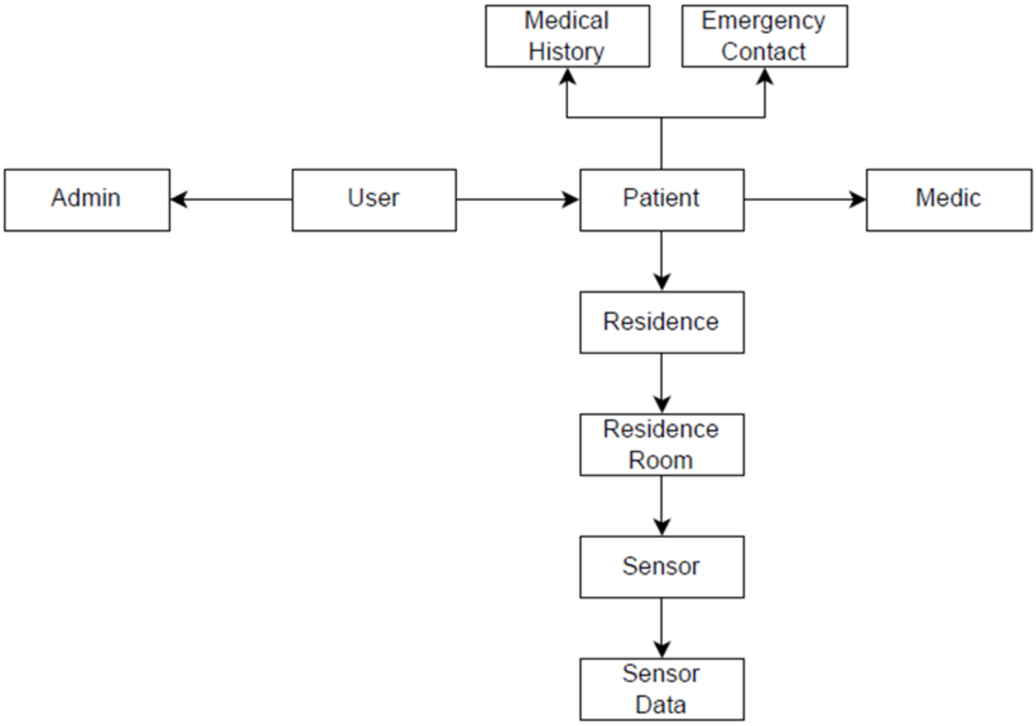


Figure 4. Alert system database logic

The user model is the default model used by the Django framework. It comes with all the features we expect such a model to have, including security, authentication, permissions and validations. When a user is created, be it a simple user or an administrator, the password is not set, and the user receives a link to a page where they can set a password on their email address.

The patient model is an extension of the user model using a unique primary key and contains basic individual information such as age, weight and height.

The residence model allows us to define the dwelling owned by the patient. It has the same restrictions as the Patient model, implementing OneToOneField to it, restricting patients to having at most one home. The model makes available a type field representing the type of residence, its values being limited to one of the values of an internally predefined list (for example, apartment, house, etc.).

The residence room model allows the management of rooms that can be defined for a Residence type entry. The information that can be filled in for a room is the type of room (bathroom, kitchen, hall, etc.) and its dimensions, with one field for each value (length, height, width).

The Sensor model allows the registration of one sensor in a room. Although it is possible to register several sensors in the same room, they are differentiated by a unique ID randomly generated and the IP address of the said sensor. The unique ID generated is formatted as two characters, followed by a sequence of six characters and another two characters, all these sequences being separated by a dash. This final code must also be recorded in the sensor that is physically installed, otherwise recording data in the database is not allowed. Furthermore, the IP of the sensor must be recorded, this being the second element without which data recording is not possible.

The SensorData model allows the data coming from the sensors to be saved in this model in JSON format. This format allows receiving data that may have other data types than expected. The JSON data should be in the following format:

```
{
"ID": "XY-XYZXYZ-XY",
"IP": "8.8.8.8",
"data": {
"status_code": &lt;int number&gt;
"datetime": &lt;datetime format&gt;
}
}
```

EmergencyContact model allows a patient to have a certain number of people (contact data) who can be contacted automatically if the system raises main alerts (falls, high temperatures, etc.).

The Medic model allows the registration of doctors in the application. If a patient is under medical supervision, this can be recorded to inform them in the event of an alert.

EnrollToMedic model has the role of assigning a doctor to a patient. Each patient can be assigned at most one doctor and a doctor can be assigned a maximum number of five patients.

MedicalHistory model can reduce the number of false-positive results, and medical data about the patient can be recorded - respectively the medical history. These can be correlated with the Sensor model data to improve performance.

4.3. Working process

The landing page of the application provides an informative summary of the status of patient and administrator accounts. A table with unset patients accounts shows information about the blank entries. When one of the lines of this table is clicked, the user is redirected to the wizard of the corresponding step, to complete the registration of the unset patient. Also, relevant information about alerts from sensors for the last 24 hours is displayed on the front page.

The application does not implement "registration" or "account creation" functions, instead, a responsible administrator can create accounts. Accounts are created with a password that cannot be used and as a result, the application sends a password-setting email.

Patient registration is a significant part of the app. Patient registration is carried out in the form of a "Wizard", in four steps, as follows:

a. Entering patients' personal data: The application creates two entries in the database to complete this step; Input in the User model and input in the Patient model. The administrator is notified on the page about any possible errors that may appear.

b. Registration of residence: The type of residence is chosen from a list.

c. At least one room/room is defined. To add the room, it is necessary to know the standard parameters of the room (length, width, height) but also the type of room (bathroom, kitchen, bedroom, etc.).

d. The last step is to define a sensor. The unique ID is generated automatically and is closely related to the physical sensor, where this ID must be registered in the sensor at the time of installation. If the sensor was installed before registration in the application, the ID can be changed to be identical. The sensor type is chosen from a predefined list (Proximity, temperature, etc.), and the model and manufacturer of the sensor are less important, but must also be registered. If several rooms were registered in the previous step, any of them can be selected at the bottom of the page where you want to register the sensor.

If problems of any kind occur while registering a patient, the administrator can always return to the section of the Wizard where they left off last time. Patient details can also be filled in from their profile pages by accessing the appropriate update section.

The registration of emergency contacts in case of an alert can be done from the user's profile page. Both patients and administrators can add, modify or delete these contacts.

A doctor can be registered in the application by an administrator. All fields are mandatory and the doctor's specialisations are relevant and consistent with the purpose of the project. It is not intended to allow doctors to access the application, but to be alerted in case of critical events.

An administrator can be registered in the application only by a user who has the is superuser field set to True. Unlike patients, not a lot of information is required, other than the essentials: username, last name, first name and email. As with patients, creating a new entry in the User table will send an email to the new user with a link that allows them to set a password.

The administration module has been designed so that all sub-components of the application are as easy as possible to populate, modify, view and analyse. It is not implemented as an administration page (for example http://telemonitor.ro/admin) but it is implemented in the graphical interface of the application and is based on access levels; In short, an administrator has more options and menus available than a normal user.

5. Conclusions

Besides using the fall detection system for a single individual, the proposed alert system can be easily used in nursing homes where dozens of old people are under care. Usually, most nursing homes have an outside yard where fall detection made by non-intrusive sensors can prove a difficult task, but this can be easily overcome by a video camera and the right fall detection algorithm as the video camera's intended purpose is security.

No system is without flaw as it is easy to identify shortcomings. Fall detection systems based on wearables raise the power supply problem but the system described in this article is not subject to this but rather to a similar problem: in case of a power failure, the system is shut down.

The alert system does not require interaction with the subject, as elderly people do not use most of today's technology. It's only required to provide some minimal information at the time of installation but this is a one-time action.

Regardless of the chosen sensor, the fall detection system can be provided as a service due to the alert system's management features.

As we have previously mentioned, real training data performed by elderly subjects are almost inexistent or insufficient to train a complex machine learning model. Most of them are performed by younger subjects who may not capture all the real-world nuances that machine learning algorithms can use in their training. The proposed software has the potential to solve this issue and gather real training data in time and provide them in the future which then can be used to better train machine learning algorithms.

Acknowledgement

Acknowledgement: This work was supported by the Ministry of Research, Innovation and Digitalization [project PN19370801 conducted at ICI Bucharest within the National Core Program].

REFERENCES

Chaudhuri, S., Thompson, H. & Demiris, G. (2014) Fall detection devices and their use with older adults: a systematic review. *Journal of Geriatric Physical Therapy*. 37(4), 178-96. doi: 10.1519/JPT.0b013e3182abe779.

Django. (2022) *Django Software Foundation*. https://www.djangoproject.com/foundation/ [Accessed: November 2022].

Igual, R., Medrano, C. & Plaza, I. (2013) Challenges, issues and trends in fall detection systems. *BioMedical Engineering OnLine*. 12, 66. doi: 10.1186/1475-925X-12-66.

Jiang, Y., Togane, M., Lu, B. & Yokoi, H. (2017) sEMG sensor using polypyrrole-coated nonwoven fabric sheet for practical control of prosthetic hand. *Frontiers in Neuroscience*. 11, 33. doi: 10.3389/fnins.2017.00033.

MDN Web Docs. (2022a) *CSS basics*. https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics [Accessed February 2023].

MDN Web Docs. (2022b) *HTML basics*. https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics [Accessed February 2023].

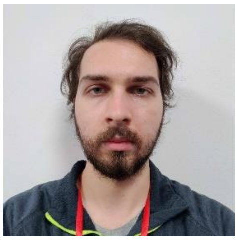
MDN Web Docs. (2022c) *JavaScript basics*. https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics [Accessed February 2023].

Usmani, S., Saboor, A., Haris, M., Khan, M. A. & Park, H. (2021) Latest Research Trends in Fall Detection and Prevention Using Machine Learning: A Systematic Review. *Sensors*. 21(15), 5134. doi: 10.3390/s21155134.

World Health Organization. (2022) *Ageing and health*. https://www.who.int/news-room/fact-sheets/detail/ageing-and-health [Accessed February 2023].

Mihai APOSTOL a absolvit Facultatea de Științe Aplicate din cadrul Universității Politehnica București, cu diplomă de inginer și a urmat un program de masterat în cadrul aceleiași instituții cu specializare în securitate cibernetică. Principalele domenii de interes și de activitate sunt dezvoltarea de aplicații web, tehnologii web si securitate cibernetică. În prezent, lucrează la Institutul Național de Cercetare-Dezvoltare în Informatică – ICI București.

Mihai APOSTOL graduated the Faculty of Applied Sciences within the Politehnica University of Bucharest, with an engineering degree and pursued a master's degree in the same institution, specializing in cyber security. The main areas of interest and activity are the development of web applications, web technologies and cyber security. Currently, he is working at the National Institute for Research and Development in Informatics – ICI Bucharest.



Răzvan-Alexandru DUȚESCU a absolvit Universitatea din Manchester (Regatul Unit), cu o diplomă de licență în Informatică, și a urmat un program de masterat în Inteligența Artificială, în cadrul Universității din Aberdeen (Regatul Unit), în 2019. Principalele lui arii de expertiză includ procesarea de imagini și învățarea automată nesupervizată. În prezent, lucrează la Institutul Național de Cercetare-Dezvoltare în Informatică – ICI București.

Răzvan-Alexandru DUȚESCU graduated from the University of Manchester (United Kingdom), with a BSc in Computer Science, and completed his MSc in Artificial Intelligence, at the University of Aberdeen (United Kingdom), in 2019. His main areas of expertise include image processing and unsupervised machine learning. He is currently working at the National Institute for Research & Development in Informatics – ICI Bucharest.