

THE APPLICATION OF MACHINE LEARNING ON THE SENSORS OF SMARTPHONES TO DETECT FALLS IN REAL-TIME

Achraf Benba¹, Mouna Akki¹, Sara Sandabad²

¹Mohammed V University in Rabat, Ecole Nationale Supérieure d'Arts et Métiers, Electronic Systems Sensors and Nanobiotechnologies, Rabat, Morocco, ²Hassan II University of Casablanca, Ecole Normale Supérieure de l'Enseignement Technique de Mohammadia, Electrical Engineering and Intelligent Systems, Casablanca, Morocco

Abstract. With the increasing prevalence of smartphones, they now come equipped with a multitude of sensors such as GPS, microphones, cameras, magnetometers, accelerators, and more, which can simplify our daily lives. When it comes to healthcare, smartphones can become indispensable. The detection of geriatric falls is crucial as even the slightest injury can have fatal consequences. Therefore, we proposed the use of accelerometers in our research to detect falls in the elderly. Our project involved the development of an automated, continuous, and reliable monitoring system that would generate a list of elderly people at risk of falling and present it on a webpage for emergency services. This approach aimed to minimize the long-term impacts and save lives promptly. We started by developing a mobile application and used MATLAB to classify the falls as either "fall" or "not fall." Finally, we created a webpage that would facilitate communication between the mobile application and MATLAB.

Keywords: fall detection, smartphone accelerometers, SVM, KNN, machine learning

ZASTOSOWANIE UCZENIA MASZYNOWEGO NA CZUJNIKACH SMARTFONÓW DO WYKRYWANIA UPADKÓW W CZASIE RZECZYWISTYM

Streszczenie. Wraz z rosnącą popularnością smartfonów są one wyposażone w wiele czujników, takich jak GPS, mikrofony, kamery, magnetometry, akceleratory i inne, które mogą uprościć nasze codzienne życie. Jeśli chodzi o opiekę zdrowotną, smartfony mogą stać się niezastąpione. Wykrywanie upadków geriatrycznych ma kluczowe znaczenie, ponieważ nawet najmniejszy uraz może mieć śmiertelne konsekwencje. Dlatego zaproponowano wykorzystanie w naszych badaniach akcelerometrów do wykrywania upadków osób starszych. Nasz projekt polegał na opracowaniu zautomatyzowanego, ciągłego i niezawodnego systemu monitoringu, który generowałby listę osób starszych zagrożonych upadkiem i prezentował ją na stronie internetowej służb ratowniczych. Podejście to miało na celu zminimalizowanie długoterminowych skutków i szybkie ratowanie życia. Rozpoczęto od opracowania aplikacji mobilnej i za pomocą MATLABa sklasyfikowano upadki jako „upadek” lub „nie upadek”. Ostatecznie stworzono stronę internetową, która ułatwiłaby komunikację między aplikacją mobilną a MATLABem.

Słowa kluczowe: detekcja upadku, akcelerometry smartfonów, SVM, KNN, uczenie maszynowe

Introduction

The World Health Organization's "OMS" reports that falls are a significant global public health issue. Falls are the second biggest cause of accidental fatality after injuries from motor accidents, with an estimated 684,000 fatalities per year. Over 80% of fatal falls have place in developing and middle-income nations. Around the world, those over 60 have the highest death rates. Even when they are not fatal, over 37.3 million falls annually are severe enough to necessitate medical attention [16].

One of the most significant risk factors for falls is age. The danger of dying or receiving serious injuries from falls increases with age for older adults. For instance, 20–30% of senior individuals who fall in the United States get moderate to serious injuries, such as contusions, hip fractures, or head damage. This level of risk might be affected by age-related physical, sensory, and cognitive changes as well as environments that are not designed for an aging population. Seniors who fall frequently get disoriented and unable to stand up or walk independently [12]. Falls can happen indoors or outside. Here, prolonged lying after a fall can result in serious injuries and is the main cause of post-accident death and chronic disorders. To stop chronic effects from occurring after a fall and to save the elderly in a timely manner, a monitoring system that is automated, continuous, and dependable is required. Therefore, it's essential to provide fall detection and remote monitoring rescue alternatives.

The most significant risk factors were a history of falls, balance and gait issues, and a number of medications. Older age, being a woman, having eye issues, diminishing cognitive function, particularly issues with attention and execution, and environmental effects are additional risk factors [8]. Smartphones have grown in importance in contemporary society as a result of their cutting-edge features, cutting-edge technology, and a variety of sensors that may be applied to the healthcare industry. Smartphones are a really portable computer that anybody can own and carry with them wherever. Anyone who falls will frequently land on the ground or another lower surface. Occasionally, a body part strikes something and arrests the fall. They stand a genuine risk of falling when they live alone or in specialized facilities, which is regrettable given how much a fall

may impair a senior's freedom. The leading cause of death for elderly individuals is falling. Over the age of 65, falls account for 80% of incidents in daily life, according to the 2016 issue of the Permanent Survey of Accidents in Everyday Life (EPAC) [1]. Our concept employs smartphone accelerometer sensors to quickly react when an elderly person falls in order to save the elderly person who lives alone in residential or hospital units.

1. Database acquisition

Any issue requiring remote monitoring of fall occurrences is covered by the provided approach [3, 9, 11]. Numerous recordings of interested subjects participating in various activity types are needed for the robust development and testing of fall detection algorithms. The SisFall dataset is selected in this instance since it provides data from 38 people (15 elderly and 23 youth). It contains diverse activity data from 19 commonplace tasks and 15 unique fall tasks, all of which were performed repeatedly during intervals of 12 to 100 seconds. SisFall data were recorded using two accelerometers, a gyroscope, and a 200 Hz sampling rate [14, 15]. In this investigation, only data from one accelerometer was used. Real-time remote monitoring of falls was done in two different scenarios: fall (F – Falls) and non fall (D – Activities of daily living ADL). We focus on the prevalence of falls among the elderly rather than the minutiae of daily routines or fall types in our work.

2. Methodology

Wearable technology is the most suitable, efficient, and aesthetically appealing solution for continuous monitoring, regardless of where the individual is or how they are positioned. Additionally, compared to ambient or camera-based systems, wearable technology is less expensive, more efficient, and less intrusive. Each person's smartphone serves as the wearable device in our approach, thus there is no additional cost.

Our project will be finished in three stages.

The three-axis accelerometer on the elderly person's smartphone is being utilized to gather information. This software sends user data and accelerometer parameters to a web page.

We created a website that would record user information and provide files for each user that contained the accelerometer's x, y, and z properties in real-time.

The website calculates the various qualities.

MATLAB emails the user repeatedly asking for these derived characteristics in order to categorize the data.

The instances (fall and not fall) are put together into a database that will be used and processed to classify the subjects based on their attributes, leading to a decision that will be delivered by MATLAB to the web page. In order to act quickly and save older people in the case of a fall, the latter compiles a registry of older people who have fallen. This stage has been completed at the level of a portable computer. The aim of this work is to investigate and create a system that can quickly detect the presence of a fall using machine learning methods. The information is continuously captured using a smartphone with a triaxial accelerometer. Our project's classes are described (fall and not fall). The organizational structure of the algorithm used in our system is shown in figures 1 and 2.

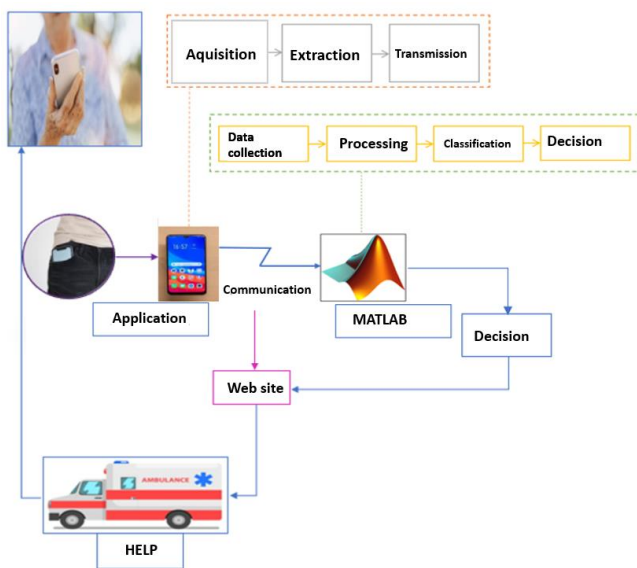


Fig. 1. Falls Remote Monitoring System Architecture

You might consider pattern recognition as a categorization technique. Their ultimate goal is to improve the extraction pattern and separate one class from another depending on certain criteria [6]. It offers the option to interpret each fresh observation in light of a body of facts or information amassed (or form). Existing observations are divided into classes in order to find new ones. It is therefore a tool for learning.

Pattern recognition is one of the many elements of artificial intelligence [10]. It makes it possible to employ a group of information or previously acquired understanding to justify each new observation (or shape). To find new observations, existing ones are grouped into classes (prototypes). It serves as a teaching tool as a result. An observation of a process is documented on a form. It is described by a collection of d parameters and represented by a point in the representation space, a space of dimension d that is specified by the different parameters (or characters).

The form vector's properties, when seen in the context of the diagnosis, represent the system under study. To associate an observed form with a recognized standard form is the issue of recognition. Typical forms (or prototypes) are representative points of this space. A fresh observation will seldom be the same as one of the prototypes due to perturbations (measurement noise, sensor precision, etc.). As a result, classes $(\omega_1, \omega_2, \dots, \omega_c, \omega_M)$ correspond to areas in space, grouping together similar forms in order to translate the impact of noise. The premise of recognition is to choose which class $[X_i = xi_1, xi_2, \dots, xi_d]$ seen should belong to out of M known classes. Classes relate to recognized modalities of operation in terms of diagnosis.

They make up the Xa learning set, which is our initial batch of data. Finding one of these modes is the first step in classifying a new observation.

Three stages are involved in the creation of a diagnostic system by RdF: perception, analysis, and operational.

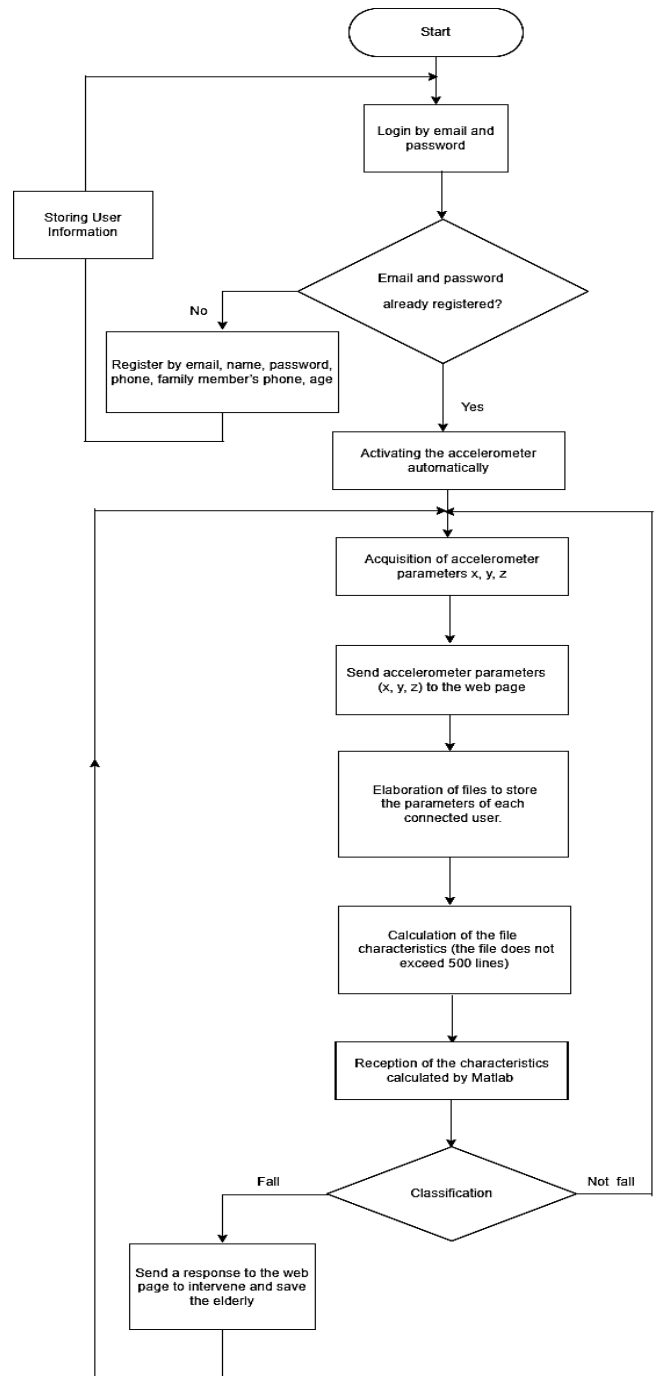


Fig. 2. Organigram of the used algorithm

The primary source of system information comes from the data collection phase. There are two steps to it. a phase in the data collecting procedure where the hardware setup necessary for signal collection on the under-review system is chosen (such as the kind, number, and sample time of the sensors to be used). The captured signals must provide pertinent data that may be used to assess the system's operational status. After this first stage, there follows a preliminary stage of signal preprocessing (filtering, de-routing, etc.). The data produced by the installed sensors in the system are examined during the analysis phase. It is essential to extract numerical features (or parameters) from the data if the information is provided as signals. These variables, which make up the form vector as well, must be able to characterize the system's behavior. At this stage of the study,

the classes that will represent the various modes of operation must be carefully defined. The following phase involves classifying a group of N observations (X_1, X_2, \dots, X_n) into M categories. This serves as the study guide. The prototypes of a class are then represented by its observations.

To divide the learning set into the several categories, a classification method is then applied. This procedure will lay out the guidelines for selecting which of the predefined classes to add a new observation to during the operational phase. In order to determine the right settings and apply the necessary changes, the analysis procedure typically entails using all of the system's data. By using the related decision rule, the operational phase (decision phase) enables the identification of an unnamed new observation X acquired on the system to one of the classes established during the classification phase. The importance of the form vector and the performance of the decision rule depend on the efficiency of the decision-making system. Before moving on to the choice, the next sections give a thorough explanation of the many stages required to create the diagnostic using pattern recognition.

At this stage, measurements of the physical system are used to produce the form vector.

The process of creating parameters from the acquired data involves applying signal processing techniques in the final phase, which is referred to as parameter generation. The best mode of operation distinction requires careful selection of these qualities.

As a result, an array of N vector forms with d parameter values, or a numerical table of size ($N \times d$), is produced from the observations. The learning package is made up of the N forms (X_1, X_N) gathered on the system.

The vector initial form's calculated parameters might not all be suitable for the investigated operating modes. Thus, it is crucial to employ parameter reduction techniques in order to maintain just the most representative parameters.

Selecting a subset of parameters that preserves the division of classes from the original learning set is the first step in reducing the form vector. Either parameter extraction methods or parameter selection methods can be used to reduce the representation space.

The unsupervised methods and supervised methods subcategories of categorization approaches are the two main groups.

When it is possible to determine the beginning class of each observation or vector form, the decision space is completely known and supervised learning is practical. The next step is to specify the allocation rules for a hypothetical form x to one of the M classes.

On the other hand, the categorization must be done in an unsupervised way if there is no information provided regarding the architecture of the classroom learning set (the observations are not tagged). When the system is not understood and it is not apparent if the measurements apply to different classes, this scenario may occur.

In this case, the observations have been used to generate classes. It entails grouping observations into classes based on shared criteria. For the rest of this study, we will only offer the supervised type categorization.

2.1. Supervised learning

The learning package was divided into several spaces at the conclusion of the analysis process. The specification of the form vector ensures that the observations that are grouped together are completely understood. Every observation in the data is labeled to indicate that it belongs to one of the recognized M classes: whatever X_i , $i = 1, \dots, N$, there is j ; $j = 1, \dots, M$ as X_i belongs to $_j$. Let X_a represent the learning set (X_1, X_2, \dots, X_N) and $(\omega_1, \omega_2, \dots, \omega_C, \omega_M)$. recognized M courses (or modes of operation in diagnostics). The next step is to decide which class to allocate a fresh X observation that was acquired at a specific time on the system. To do this, decision boundaries between classes must be constructed together with a decision procedure.

The decision methods that were used define a categorization rule for new observations to the various classes of the learning set. The decision rule might be created by the use of statistical or analytical techniques.

2.1.1. Using statistical analysis: KNN

A data categorization technique known as the k -Nearest Neighbor (KNN) algorithm calculates the likelihood that a data point will belong to one group or another depending on the group to which the data point nearest to it belongs [7, 17]. Following are the stages involved in classification using KNN.

Select the K whose categorization will work best.

The best prediction rate is achieved when K is between 5 and 18; above this amount, we can witness the phenomena of "overfitting", which happens when a model becomes very familiar with the specifics and noise of training data to the point that it degrades model performance on fresh data. Following are the stages involved in classification using KNN.

The Euclidean distance between two points in Euclidean space is the length of the right segment that connects them. The Pythagorean theorem may be used to calculate it from the points' Cartesian coordinates, thus the name Pythagorean distance.

2.1.2. SVM as the analytical method

This approach looks for the mathematical representations of the boundaries that best define classes using data from the learning set. Which boundary is most suited depends on the complexity of the decision boundary, or more precisely, on whether or not classes are properly split from the learning set. The goal of the SVM approach is to locate a hyperplane in an N -dimensional space ($N =$ number of features) that classifies data points in a certain manner [2]. These two types of data points can be split using a variety of possible hyperplanes. We are seeking for the greatest gap between two different data point types at level H , which has the broadest margin. The advantage from increasing edge distance helps subsequent data points be identified more accurately. The data points at the slab's edge that are closest to the separation hyperplane are known as support vectors. These definitions are shown in the accompanying picture, where the $+$ denotes Type 1 data points and the $-$ denotes Type -1 data points.

3. Results

When we examine at the first three columns of the database's accelerometer 1, which are seen in figure 3, we see that there are many entries for each individual.

Fichier	Edition	Format	Affichage	Aide
-6,-248,	2,	-57, 108,	-49,	-34,-963, 151;
-8,-247,	7,	-53, 106,	-47,	-32,-966, 157;
-4,-247,	2,	-49, 106,	-43,	-34,-966, 162;
-8,-247,	10,	-44, 105,	-39,	-31,-963, 162;
-6,-245,	10,	-40, 104,	-38,	-34,-964, 165;
-9,-242,	6,	-36, 102,	-36,	-33,-964, 164;
-8,-245,	8,	-34, 100,	-34,	-29,-960, 166;
-4,-246,	11,	-34, 99,	-33,	-30,-961, 165;
-5,-244,	13,	-37, 97,	-33,	-31,-956, 168;
-5,-244,	12,	-40, 95,	-32,	-32,-956, 169;
-4,-246,	8,	-43, 94,	-31,	-26,-955, 163;
-9,-246,	12,	-48, 96,	-31,	-30,-956, 175;
-8,-244,	12,	-48, 97,	-30,	-34,-955, 171;
-9,-245,	13,	-51, 100,	-31,	-33,-960, 166;
-8,-249,	8,	-53, 101,	-32,	-37,-957, 167;
-8,-253,	10,	-52, 103,	-34,	-32,-963, 170;
-9,-245,	13,	-54, 104,	-34,	-35,-961, 171;
-10,-247,	10,	-56, 104,	-37,	-35,-967, 170;
-10,-247,	12,	-57, 107,	-39,	-37,-962, 169;
-9,-244,	12,	-55, 108,	-43,	-35,-965, 169;
-7,-246,	9,	-57, 110,	-45,	-30,-967, 168;
-9,-247,	11,	-57, 112,	-47,	-32,-965, 165;

Fig. 3. Signals of first accelerometer from database

These values are then normalized by the following equation:

$$Num = \left(\frac{2 \times Range}{2 \times Resolution} \right) \times Num \quad (1)$$

with: Range = 16; Resolution = 13.

The following parameters were calculated from each axis of the first accelerometer and from each record in the next step:

Maximum peak to peak amplitude, Maximum, Variance, Standard Deviation, and Minimum.

A confusion matrix resembles a list of forecasts for a certain categorization issue. It contrasts the target variable's actual data with the model's predictions for that variable.

There are 4500 items in our database, including 2702 from the activities of daily living (ADL) class and 1798 from the falling scenario class. The strategy of holdout validation was used. A total of 3375 records, or 75% of the database, were used in the training phase. 1349 recordings were made in a falling scenario, while 2026 recordings were made in an ADL scenario. For the validation phase, we used a total of 1125 recordings, which accounts for the remaining 25%. 676 recordings from the ADL situation and 449 recordings from the fall semester.

In order to compute performances of our algorithms, several methods were used:

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \times 100 \quad (2)$$

$$Sensitivity = \frac{T_P}{T_P + F_N} \times 100 \quad (3)$$

$$Precision = \frac{T_P}{T_P + F_P} \times 100 \quad (4)$$

$$Specificity = \frac{T_N}{T_N + F_P} \times 100 \quad (5)$$

$$F1 - Score = \frac{2 \times Precision \times Sensibility}{Precision + Sensibility} \quad (6)$$

As seen in figure 4, the confusion matrix shows what happens when the best kernel SVM model is used. As we can see, only 435 out of the fall instances were accurately categorized, whereas 14 of them were. Out of 667 recordings that were accurately identified for the ADL scenarios, only 9 recordings were misclassified.

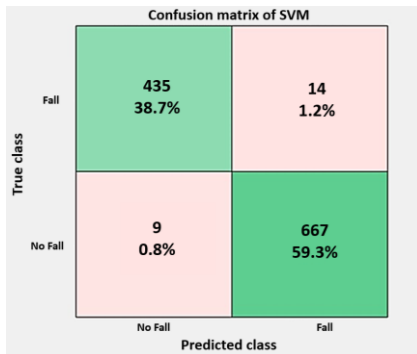


Fig. 4. SVM confusion matrix

The confusion matrix in figure 5 displays the results obtained using the KNN model with the best closest neighbor. As we can see, just 3 fall scenario classifications were inaccurate, leaving 446 properly classified. Out of 675 recordings that were successfully categorized for the ADL scenarios, just one was misclassified.

Table 1 lists the outcomes from the SVM and KNN models. As seen in table 2, both models' accuracy is highly outstanding, however the KNN model's accuracy score is significantly greater at over 99.6%. The KNN model also performs better in terms of sensitivity, accuracy, specificity, and the F1-score. Tables 3 and 4, which summarized the findings, show that KNN and SVM models were flawless in their ability to anticipate every situation.

Table 1. Obtained results using hold out validation scheme for KNN and SVM models

Methods	SVM	KNN
True Positive TP	435	446
True Negative TN	667	675
False Positive FP	9	1
False Negative FN	14	3

4. Discussions

For the fall situations and ADL scenarios in the aforementioned results, there were several recordings for each individual. We calculated the averages of the retrieved variables for each participant in order to obtain only one recording for each one. There are already 62 recordings in our new database, along with 24 falls situations and 38 ADL scenarios. We utilized 75% of the database for training and 25% for testing. Figure 6 shows the confusion matrix for the SVM model and Figure 7 shows the confusion matrix for the KNN model.

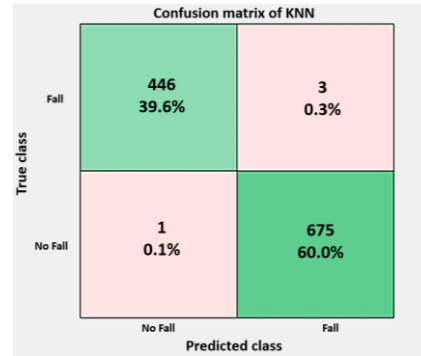


Fig. 5. KNN confusion matrix

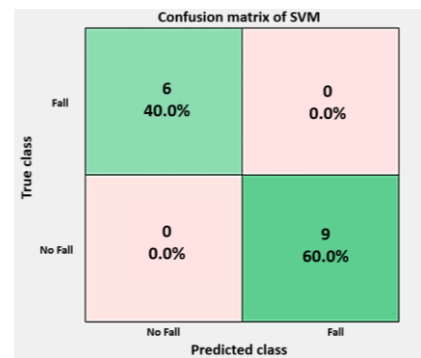


Fig. 6. SVM confusion matrix after computing the average

Table 2. Performance comparison between SVM and KNN

Parameter	SVM	KNN
Accuracy	97.9%	99.6%
Sensitivity	96.8%	99.3%
Precision	97.9%	99.7%
Specificity	98.6%	99.8%
F1-Score	97.34%	99.49%

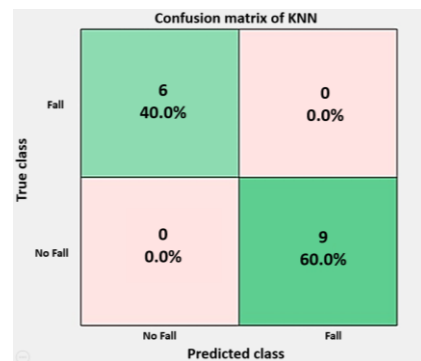


Fig. 7. KNN confusion matrix after computing the average

We used the Google-first, Massachusetts Institute of Technology-run MIT App Inventor [13], an online tool for developing Android apps for smartphones and tablets (MIT).

Three windows are available for development:

- One for designing the user interface for a machine; this is how your software will appear.
- One for self-programming, which enables the construction of application behavior out of building pieces.

Provides a testing environment for the software. The emulator makes it possible to verify the proper execution of the application rather than using the terminal.

Table 3. Obtained results using hold out validation scheme for KNN and SVM models after computing the average

Methods	SVM	KNN
True Positive TP	6	6
True Negative TN	9	9
False Positive FP	0	0
False Negative FN	0	0

Table 4. Performance comparison between SVM and KNN after computing the average

Parameter	SVM	KNN
Accuracy	100%	100%
Sensitivity	100%	100%
Precision	100%	100%
Specificity	100%	100%
F1-Score	100%	100%

We can download the application and test it out by connecting to a real Android smartphone. The application will work the same way whether this terminal is a phone or a tablet.

We'll develop a program that captures user motion, collects information from the accelerometer's x, y, and z axes, and transmits that information to a website we have developed to look for signs of a fall.

To create the interface, we picked each component, moved it into the phone's screen, and adjusted its options:

- Screen 1 is the application's main screen; it is shown for 5 seconds before switching to screen 2.
- Using the technique shown in figure 8, the user may sign up or log in using screen 2 (email and password).
- Screen 3 enables user registration (email, name, passwords, phone number, phone number of a family member and age).
- If the criteria is not verified, screen 4 enables you to immediately activate the accelerometer, display the accelerometer parameters in the program interface, and communicate these information to the webpage in real time.

For this job, we developed a dynamic website. A database, which is a collection of structured data that enables research, maintenance, and updating [4], may be handled by our website utilizing MySQL and PhpMyAdmin. Data is arranged in rows, columns, and tables in databases. They are indexed so that using computer software, the required information may be found fast. Data is updated and erasable when further information is supplied.

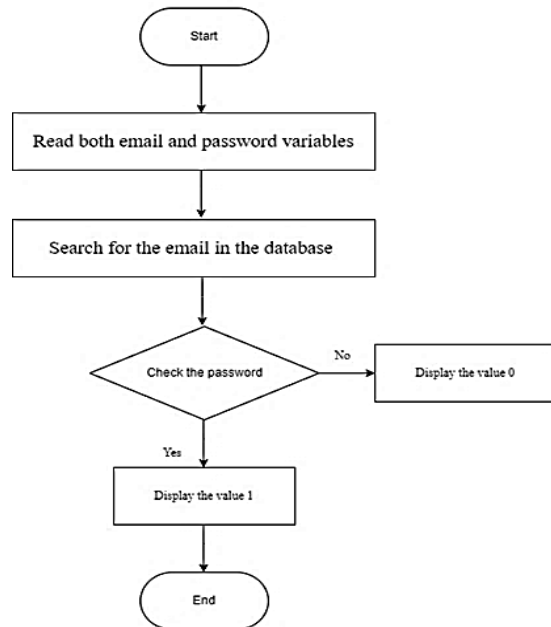
The smartphone may be used in any direction; there is no specific position in which to utilize it for precise results. We tested our system in a seemingly random location, and it worked well.

Figures 9, 10, 11, and 12 depict the many steps of communication between the application, website, and MATLAB for each level.

For instance, the user line and the -connect- column box both display 1 when the user (mouna@gmail.com) inputs their email address and password to access the website. The webpage creates a file with the email and accelerometer parameters that is no longer than 500 lines long (if it is, the first 500 lines are erased to maintain the file at 500 lines), then calculates the characteristics. Given that 200 ms is the greatest amount of time needed for an old person who is falling to strike the ground, the rate of accelerometer data receipt should be less than 200 ms [5]. Therefore, even if there is a delay in transmission,

it is imperative to continuously send all signal samples, and the entire history should be saved and delivered. Although we analyze 500 lines in our solution, we may run additional studies to account for all delayed samples. The accelerometer activates automatically (Minimum, Maximum, Average, Median, Variance, Standard deviation, Maximum amplitude of peak to peak). MATLAB will repeatedly ask for these characteristics in order to categorize the data and send the classifications to the website.

Log in



Sign up

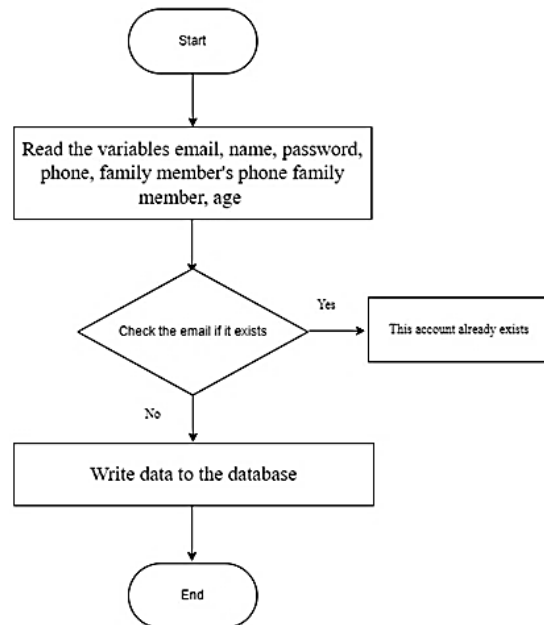


Fig. 8. Flowchart of registration and connection

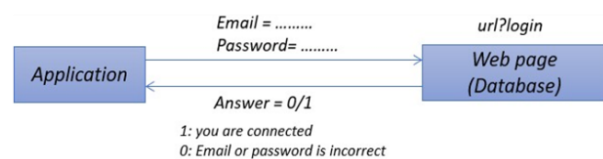


Fig. 9. Login phase

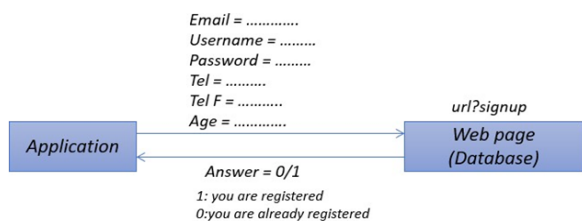


Fig. 10. Sing up phase

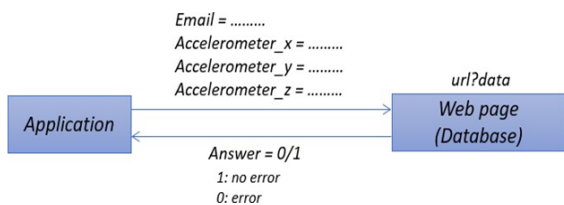


Fig. 11. Data recording phase

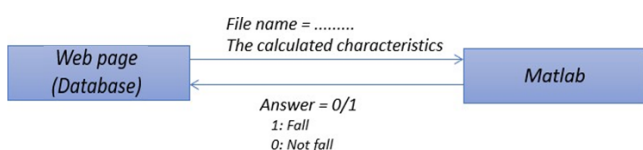


Fig. 12. State assessment phase

In order to examine and save older individuals or anybody else when a fall happens, a web portal that enables the emergency services to link and visualize each person who requires examination is required. We have created a website just for emergency services (username and password must be distinct and recognized by all emergency services).

To provide a summary of the entire project, let's assume that this is the case. When a new user registers and connects, the accelerometer is automatically activated and the parameters are sent to the web page. A file is created for the new user once the website examines the properties. MATLAB repeatedly asks the 4 users (the new user plus the users who are already connected) to do the classification in real time. After ascertaining that the newly connected user and the presently connected users are both falling, MATLAB allocates 1 to the "falling" box of the four users. The website then compiles a list of topics that are losing popularity.

5. Conclusion

Even while a fall could be unpleasant, it could be lethal for someone who has certain conditions. for example, elderly people or those who experience seizures. Therefore, precise and straightforward fall detection is crucial and can help in protecting and assessing these kinds of persons. We choose smartphones since they are actually accessible to everyone, offer commuting interfaces, and include sensors and GPS localization in addition to other features. As we demonstrated, the system we developed distinguishes between fall situations and other scenarios involving everyday tasks more precisely. A web service, a processing node that in our case utilizes MATLAB to make choices, and a mobile app that can be utilized on any device make up our system. The emergency center in charge of assessing these persons is also alerted when a fall happens. Overall, we believe that this approach will be quite helpful, and the results are both hopeful and pleasant. By enabling it to sleep when no activity is being detected and lowering the frequency of data collection, this program's energy-saving features might be improved. In the presence of activity or motion, data collection frequency must rise.

Conflict of interest

No author claims to have any conflicts of interest.

References

- [1] Bouilly M., Thélot B.: Les accidents de la vie courante aux urgences en France métropolitaine en 2010 selon l'enquête EPAC. Revue d'Epidémiologie et de Santé Publique 60, 2012, S145.
- [2] Cherkassky V., Ma Y.: Practical selection of SVM parameters and noise estimation for SVM regression. Neural networks 17(1), 2004, 113–126 [http://doi.org/10.1016/S0893-6080(03)00169-2].
- [3] El Attaoui A. et al.: Machine learning-based edge-computing on a multi-level architecture of WSN and IoT for real-time fall detection. IET Wireless System Systems 10(6), 2020, 320–332 [http://doi.org/10.1049/iet-wss.2020.0091].
- [4] Enterprise J.: HTML, PHP, dan MySQL untuk Pemula. Elex Media Komputindo 2018.
- [5] Er P. V., Tan K. K.: Non-intrusive fall detection monitoring for the elderly based on fuzzy logic. Measurement 124, 2018, 91–102 [http://doi.org/10.1016/j.measurement.2018.04.009].
- [6] Fukunaga K.: Introduction to statistical pattern recognition. Elsevier 2013.
- [7] Guo G. et al.: KNN model-based approach in classification. Lecture Notes in Computer Science 2888, 2003 [http://doi.org/10.1007/978-3-540-39964-3_62].
- [8] James K.: Falls and Fall Prevention in the Elderly. West Indian Med J. 56(6), 2007, 534.
- [9] Le H. L. et al.: A novel feature set extraction based on accelerometer sensor data for improving the fall detection system. Electronics 11(7), 2022, 1030 [http://doi.org/10.3390/electronics11071030].
- [10] Li D., Wu M.: Pattern recognition receptors in health and diseases. Signal transduction and targeted therapy 6(1), 2021, 291 [http://doi.org/10.1038/s41392-021-00687-0].
- [11] Noury N. et al.: Fall detection-principles and methods. 29th annual international conference of the IEEE engineering in medicine and biology society, 2007, 1663–1666 [http://doi.org/10.1109/IEMBS.2007.4352627].
- [12] Pannurat N. et al.: Automatic fall monitoring: A review. Sensors 14(7), 2014, 12900–12936 [http://doi.org/10.3390/s140712900].
- [13] Patton E. W.: MIT app inventor: Objectives, design, and development. Computational thinking education, 2019, 31–49 [http://doi.org/10.1007/978-981-13-6528-7].
- [14] Rashid F. A.: Simulation of SisFall Dataset for Fall Detection Using MATLAB Classifier Algorithms. 12th International Symposium on Parallel Architectures, Algorithms and Programming – PAAP, 2021, 62–68, [http://doi.org/10.1109/PAAP54281.2021.9720481].
- [15] Sucerquia A. et al.: SisFall: A fall and movement dataset. Sensors 17(1), 2017, 198 [http://doi.org/10.3390/s17010198].
- [16] World Health Organization: Ageing, Life Course Unit. WHO global report on falls prevention in older age. World Health Organization; 2008.
- [17] Zhang S. et al.: Learning k for kNN classification. ACM Transactions on Intelligent Systems and Technology 8(3), 2017, 1–9 [http://doi.org/10.1145/2990508].

Prof. Achraf Benba
e-mail: achraf.benba@ensam.um5.ac.ma

Received the Ph.D. degree in electrical engineering from ENSAM, Mohammed V University, Rabat, Morocco, in 2017. He is a member of E2SN teams at ENSAM, Mohammed V University. His interests are in speech processing for detecting people with neurological disorders and Cardiac abnormalities.



<http://orcid.org/0000-0001-7939-0790>

Eng. Mouna Akki
e-mail: mouna.akk72@gmail.com

Received the engineering degree in biomedical engineering from ENSAM, Mohammed V University, Rabat, Morocco, in 2022. She is interested in embedded systems applied in health assessment.



<http://orcid.org/0009-0001-8532-1158>

Prof. Sara Sandabad
e-mail: s.sandabad@isem.ac.ma

Received the B.Sc. and the M.Sc. degrees in Electrical Engineering from ENSET, Mohamed V University of Rabat (Morocco) and the Ph.D. degree from ENSIAS from the Mohamed V University of Rabat, all in Electrical Engineering, in 2010, 2012, and 2016, respectively. Currently, she is a Professor of Electrical Engineering at ISEM, Hassan II University of Casablanca. She is also an academic member of Laboratory of Control and Mechanical Characterization of Materials and Structures, ENSEM, Hassan II University of Casablanca, Morocco.



<http://orcid.org/0000-0002-0813-6178>