

# Accelerometer and Camera-Based Strategy for Improved Human Fall Detection

Nabil Zerrouki<sup>1</sup> · Fouzi Harrou<sup>2</sup> · Ying Sun<sup>2</sup> · Amrane Houacine<sup>1</sup>

Received: 26 May 2016 / Accepted: 14 October 2016 / Published online: 29 October 2016  
© Springer Science+Business Media New York 2016

**Abstract** In this paper, we address the problem of detecting human falls using anomaly detection. Detection and classification of falls are based on accelerometric data and variations in human silhouette shape. First, we use the exponentially weighted moving average (EWMA) monitoring scheme to detect a potential fall in the accelerometric data. We used an EWMA to identify features that correspond with a particular type of fall allowing us to classify falls. Only features corresponding with detected falls were used in the classification phase. A benefit of using a subset of the original data to design classification models minimizes training time and simplifies models. Based on features corresponding to detected falls, we used the support vector machine (SVM) algorithm to distinguish between true falls and fall-like events. We apply this strategy to the publicly available fall detection databases from the university of Rzeszow's. Results indicated that our strategy accurately detected and classified fall events, suggesting its potential application to early alert mechanisms in the event of fall situations and its capability for classification of detected falls. Comparison of

the classification results using the EWMA-based SVM classifier method with those achieved using three commonly used machine learning classifiers, neural network, K-nearest neighbor and naïve Bayes, proved our model superior.

**Keywords** Fall detection and classification · Visiosurveillance · Tri axial accelerometer · Anomaly detection · Support vector machine

## Introduction

## Background

The direction of activity recognition research is receiving increasing attention, motivated by the rapid development of information, communication technologies and intelligent video systems. This paper presents a method of human fall detection aimed at alleviating this major health concern. Several studies have shown that falls are a main reason for the hospitalizations of elderly, disabled, overweight and obese people [1, 2]. As shown in the study [3] given by the World Health Organization, 30 % of people over 65 fall at least one each year [3], and that 47 % of those who have fallen cannot get up without help [4]. Furthermore, a Eurostat study predicted that this category of the population will increase from 17,1 % to 30 %. This represents a rise from 84.6 million people in 2008 to 151.1 million people in 2060 [3]. Although often not fatal, falls can lead to serious injury and even death [5]. Furthermore, effects on health are more detrimental if the person remains lying on the ground for a long period of time after falling (long-lie). By 2020, falls are predicted to increase medical care expenditures by \$43.8 billion [6, 7]. Thus reliable fall detection and classification systems will improve quality of life, and increase

---

This article is part of the Topical Collection on *Systems-Level Quality Improvement*.

---

✉ Fouzi Harrou  
fouzi.harrou@kaust.edu.sa

<sup>1</sup> LCPTS, Faculty of Electronics and Computer Science, University of Sciences and Technology Houari Boumédiène (USTHB), Algiers, Algeria

<sup>2</sup> Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division, King Abdullah University of Science and Technology (KAUST), Thuwal, 23955-6900, Saudi Arabia

levels of safety and potentially reduce medical expenses [7, 8].

Numerous networks and research projects (e.g., E-NO FALLS network, Fall Detector For The Elderly (FATE), the European profound project, Fall-Watch and BIOTELEKINESY) have been launched in an effort to design reliable, unobtrusive fall detection models for prevention and increase safety aimed at improving the quality of life of the elderly. Over the past two decades, researchers and engineers have developed several fall detection techniques that can be split into two main approaches: non computer vision based and computer vision based [9–17]. Non computer-vision-based fall detection approaches typically depend on information captured by sensors. These methods use acoustic sensors [18], floor vibration sensors [18] and human body movements to detect a fall [15–17, 19, 20]. However, the reliability of acoustic sensors is limited by their sensitivity to extraneous noise, causing a high number of false alarms, floor vibration sensors can only provide information about surface equipped with sensors. Installation of floor vibration sensors is costly if they cover much area and they are only effective if they cover the location of a fall. Bodily movements can be detected by wearable sensors (e.g., inertial measurement units), which are emerging as a favorable approach [10, 21, 22] because they are easy to set up, low cost [21] and perform automatic fall detection [9, 23–25]. Wearable sensors-based fall detection can be categorized into two main classes: i) threshold-based techniques, in which a fall is declared when acceleration magnitude overpass a predefined threshold; and (ii) machine-learning techniques [26]. Bourke et al. [27] used a bi axial gyroscope, where threshold and acceleration angles on a chest are defined to identify a fall. Sposaro et al. presented a fall detection solution called iFall, which uses sensors embedded on smart watches and smartphones [28]. False alarms ratio can be reduced by applying a verification phase, where once the fall is detected, a verification message is sent asking the user's confirmation. If the user doesn't reply, an alert message is automatically sent. Tong et al. proposed a human fall detection and prediction system using a Hidden Markov model classifier, where tri axial accelerometer data is collected as a time series [22], and Gibson et al. proposed a multi-classifier system for accelerometer-based fall detection [19].

Alternatively, computer-vision-based fall detection methods rely on information, such as shape, obtained from images and videos [9, 20, 29]. Various vision systems have been investigated, such as single charge-coupled device camera [29], multiple cameras [30], specialized omnidirectional cameras [31] and stereo-pair cameras [32]. Liu et al. [33] and Kwolek [20] proposed using the body's center of gravity as an indicator of a human fall by evaluating if the distance between two centers on two adjacent frames

is significantly different [33] or if the body's center of gravity is too close to the floor [20]. These approaches have limited application because the floor can be assimilated into background, making it difficult to differentiate it from the monitored person in each frame [20]. Rougier et al. [28] extracted the body's change in shape information and the head's velocity to manually appropriate a threshold to distinguishing between fall-like and true-fall sequences. However, using these methods, a high false detection ratio is obtained (several fast sitting activities were misclassified as a falls) and its performance was strongly dependent on the threshold value. Auvinet et al. [10] proposed fall detection using a reconstructed 3-dimensional human silhouette. Falls were detected with reasonable accuracy by considering the volume distribution along the vertical axis, such that an alarm was generated when the majority of this distribution was suddenly near the floor according the predefined time threshold. However, this method required many cameras (four or more), and a graphic processing unit was necessary for treatment. Foroughi et al. [34] proposed using an approximated ellipse of the silhouette of a human body as features for developing a neural network classifier to identify falls. Kwolek and Kepski [10] presented a fall detection method using a depth camera and sensor data; falls were classified by support vector machine (SVM) algorithm. Similarly, Bian et al. [35] presented joint extraction features based on a pose-invariant randomized decision tree; a support vector machine classifier was used to distinguish between falls and daily activities. Stone [36] characterized a body's vertical state in frame sequences, and an ensemble of decision trees was applied to determine the confidence that a fall had really occurred.

### Motivation and contribution

To achieve improved performance of fall detection and classification, we use both visual data from a camera and accelerometric data captured by an accelerometer. We start by using only accelerometric data in the detection stage to provide real-time, and fast processing; these data do not require any pretreatment. We use an accelerometer as the fall indicator in this study. To reduce false alarms caused by confusion between real falls and fall-like activities, a classification stage is applied only to detected cases through visual data. The main contributions of this paper can be summarized as follows:

- Indeed, most existing fall detection methods based on accelerometer data compare the amplitude of acceleration with a prefixed threshold that is manually fixed and generally results in a high number of false alarms. The methods described in this paper, aim to reduce errors in detecting fall events using a statistical framework

for successful early fall detection. We were motivated by the superiority of the exponentially weighted moving average (EWMA) metric to detect small changes in the process mean [37], making it very attractive as an anomaly detection chart. The main advantage of this method is that it can easily be implemented in real time because of its low computational cost. Specifically, we use the EWMA control chart based on the accelerometric data as a starting point to reduce the size of the data used as input in the classification phase. This considerably reduces computational burden and favors reasonable accuracy. Then, we employ a classification algorithm to further classify detected falls.

- In fact, only data from the camera is concerned in the classification step. In this work, we define features that will successfully differentiate real falls from fall-like events. We base the extracted features on non-zero pixels that constitute the human body. More specifically, we use five partial occupancy areas (i.e., the number of pixels in each area of an image) of the body to detect and classify falls. Only features corresponding to detected falls will be used in the classification phase. Using a subset of the original data to design classification models has the benefits of minimizing training time and promoting model simplicity. Once an event was classified as a fall based on these features, we applied the SVM algorithm to verify the event as a true fall. We chose the SVM classifier for its high generalization performance; its algorithm can behave as a linear or nonlinear classifier by exploiting linear or nonlinear kernels, respectively. And because SVM maps the input vector (feature vector) into a higher feature space using the kernel trick, it can solve linearly non separable cases. This is particularly helpful when features are confused, because these features are non linear.

The rest of the paper is organized as follows: Section “Fall detection” presents fall detection based on accelerometric data and outlines the EWMA monitoring chart and its use in fall detection. Section “Classification strategy” presents the vision-based fall classification steps and the classification algorithms used to distinguish real falls from fall-like events. Section “Experimental results” evaluates the performance of the proposed method, and Section “Conclusion” concludes this study with some remarks.

## Fall detection

In this paper, we propose fall detection strategy that comprises two main complementary phases: detection and classification. The detection phase uses statistical monitoring charts to detect falls based on accelerometer data in real-time. Once the fall is detected, its type is identified (real

fall or fall-like). However, detection based only on inertial sensors presents some limitation in separation between real falls and some ordinary activities (generally considered as sudden movements). To discern real falls from fall-like events, we used monitoring charts that identify features corresponding to falls that are useful for fall classification purposes. Furthermore, the key role of the classification phase is to distinguish between a real fall, and activities similar to a fall such as intentional lying down or stretching out. Specifically, in this phase only the detected sequences are used as the input data for classification, making this method computationally fast. This classification method separates sequences into real fall and fall-like classes by performing an efficient background subtraction algorithm to extract the human body foreground. To improve the segmented images, a post-processing stage is applied. Features describing human posture are then extracted and used as the input for classification purposes. When a sequence is classified as a fall, an alarm signal is sent to caregivers (such as emergency or family members) via modern communication systems. The flowchart of the proposed fall detection system is illustrated in Fig. 1. Each step will be described in detail in the next sections. In this section, the detection phase will be described.

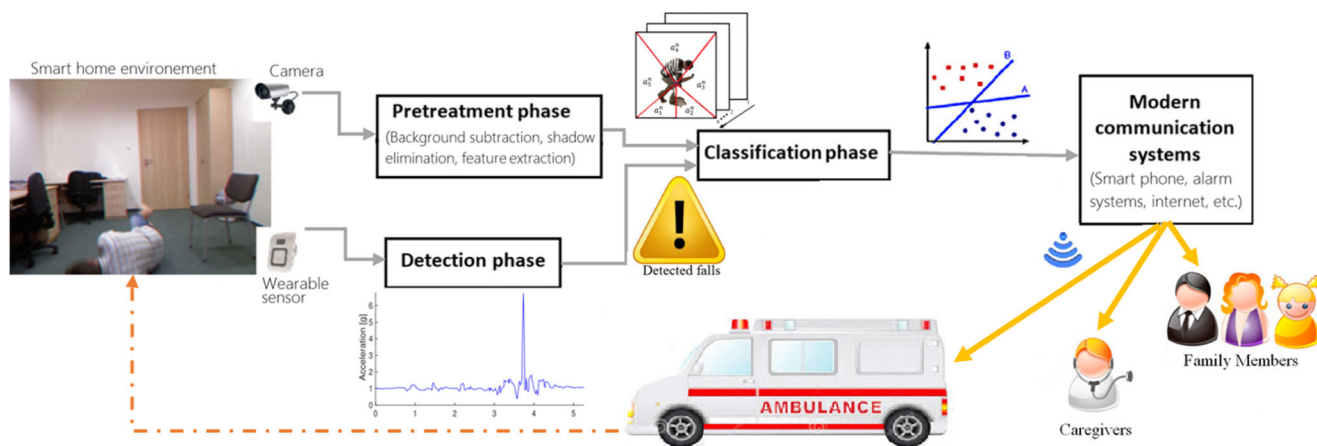
## Fall detection using accelerometric data

The binary decision making process of fall detection requires that falls be identified from non-fall events based on some relevant features of the data. In this work, we present fall detection algorithms based on accelerometer data taken from the University of Rzeszow’s fall detection dataset (URFD), a set of publicly available acceleration data collected via an accelerometer x-IMU (256Hz) [10]. The placement of the sensors is important because different parts of the body accelerate at different rates during movements. As reported in [10, 22, 38], for an accelerometer it should be placed in the pelvis region to properly represent the movements involved in a fall because this region provides the best approximation of the mass center [10]. Thus, in this study we also placed the IMU accelerometer near the pelvis region, as illustrated in Fig. 2. The accelerometer collects data in three dimensions: longitudinal acceleration ( $x$ -axis), lateral acceleration ( $y$ -axis) and vertical acceleration ( $z$ -axis). Here, the magnitude of acceleration,  $a$ , in three-dimensional space, which is the vector norm, is used as fall indicator and is defined as

$$a = \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)}, \quad (1)$$

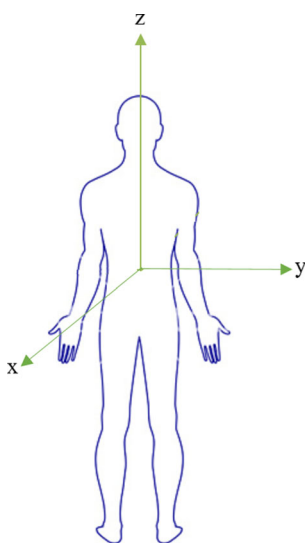
where  $a_x(t)$ ,  $a_y(t)$ ,  $a_z(t)$  represent the acceleration components  $x$ ,  $y$ ,  $z$  axes, respectively, at the instant  $t$ .

The magnitude of acceleration is discriminative enough to describe the human activities because both dynamic and



**Fig. 1** Schematic representation of a fall detection system

static accelerations are considered [10, 21, 22]. The peaks in acceleration magnitude, which are generated by the ground reaction force when the foot hits the ground, are commonly used to detect a fall [39]. Examples of acceleration change curves during daily activities like picking up an object, sitting down standing up and a fall event are depicted in the top of Fig. 3 from left to right. The plots shown in the bottom of Fig. 3 illustrate the corresponding change of in tri-axial acceleration components. It can be observed that during sitting or down or standing, the value of  $a$  is around  $1.7g$ , during some daily activities, such as picking up an object approximately  $2.7g$  and during fall  $6.8g$ . Note that the magnitude of acceleration changes significantly when falls occur in comparison to non-fall activities (Fig. 3). Using this comparison operation as an indicator, we allow for a relatively low complexity of computation for fast processing as a fall detection method.

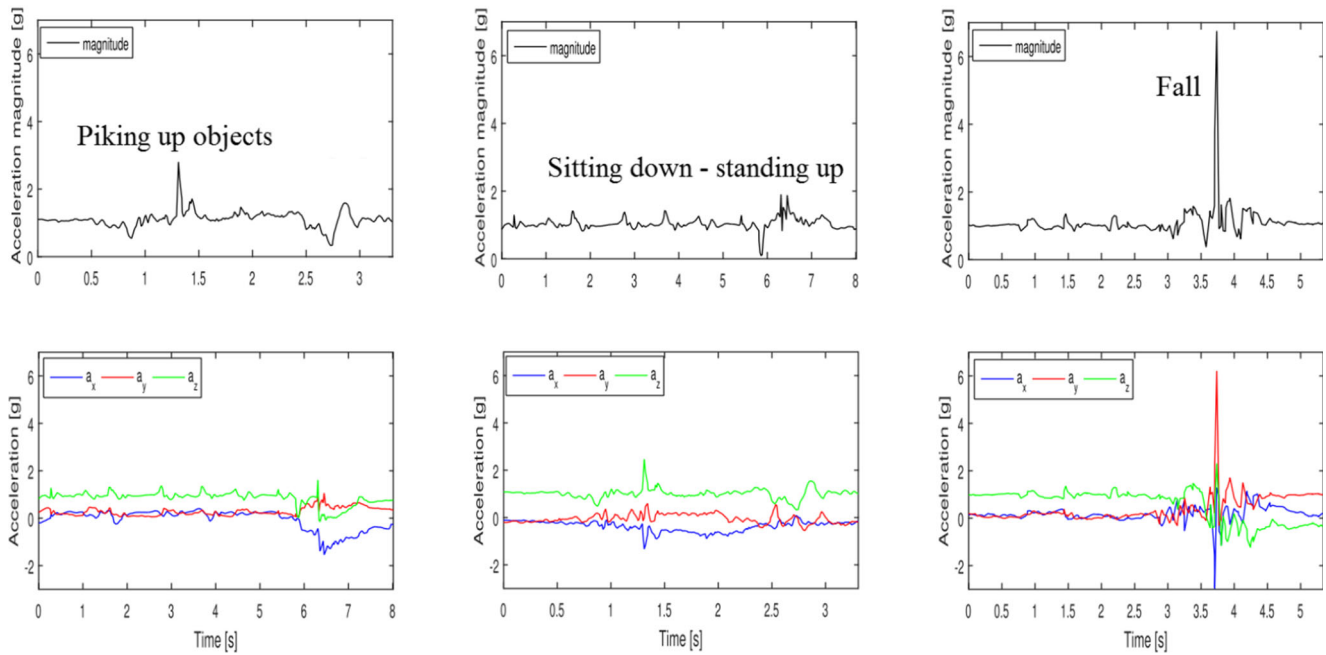


**Fig. 2** Schematic of a coordinate system

One common fall detection approach using tri-axial accelerometer data functions by defining a threshold for triggering an alarm: a fall is declared when the magnitude of acceleration surpasses this prefixed threshold value. The threshold value, however, is fixed manually based on the magnitude of acceleration and thus cause a considerable number of false alarms. Other machine-learning algorithms have been used to detect falls but they typically have relatively high computational cost because their estimation and tuning model parameters cannot be used in real time. Here, we lower computational costs of detecting falls using the EWMA algorithm. Univariate statistical monitoring methods, such as the EWMA control scheme [40], have only recently been introduced to detect small anomalies and improve performance of the human fall detection. Here, our main goal is to exploit the advantages of the EWMA monitoring scheme to detect a fall based on the magnitude of acceleration. We present a brief introduction to EWMA and how it can be used in fall detection in the next Section.

**Exponentially weighted moving average monitoring chart**

Detecting the particular anomalies that occur in a monitored system is based on checking whether the current measurements are statistically different from the a priori known faultless measurements (i.e., measurements without anomalies). EWMA charts weight the last observation based on its importance in characterizing the process [41]. These charts can detect small shifts in the process mean as they are time-weighted. According to the literature, EWMA is one of the most frequently used control charts for monitoring process mean because of its flexibility (allows suitable parameters to be selected to achieve the highest possible performance) and sensitivity to small shifts. The EWMA control scheme was first introduced by Roberts [42], and it has since been extensively used in time series analysis [37, 40, 43–46]. Assume



**Fig. 3** Acceleration (*top row*) and the tri-axial acceleration components (*bottom row*) observed while, (*left panel*) sitting down standing up, (*middle panel*) picking up an object and (*right panel*) falling

that  $\{x_1, x_2, \dots, x_n\}$  are individual observations collected from a monitored process. The EWMA statistic is computed using the following equation [40]:

$$\begin{cases} z_t = \lambda x_t + (1 - \lambda) z_{t-1} & \text{if } t > 0 \\ z_0 = \mu_0, & \text{if } t = 0, \end{cases} \quad (2)$$

where  $\mu_0$  represents the mean of fault-free data and  $\lambda \in (0, 1]$  is a weighting parameter that determines the temporal memory of the EWMA decision function. The EWMA chart is generally suitable for detecting relatively small changes when  $\lambda$  is chosen to be small, and it is suitable for detecting relatively large changes when  $\lambda$  is chosen to be large [40, 47]. The literature recommends using a smoothing parameter value between 0.2 and 0.3 for most monitoring purposes [40, 48]. This chart provides us with significant evidence to conclude that there is an anomaly at the  $i$ -th time point if the decision statistic  $z_t$  exceeds the control limits that are expressed in terms of the asymptotic standard deviation of the decision statistic.

$$z_t < \mu_0 - L\sigma_{z_t} \text{ or } z_t > \mu_0 + L\sigma_{z_t}, \quad (3)$$

where  $\sigma_{z_t} = \sigma_0 \sqrt{\frac{\gamma}{(2-\gamma)} [1 - (1-\gamma)^{2t}]}$  is the standard deviation of  $z_t$ ,  $\sigma_0$  is the standard deviation of the anomaly-free or preliminary data set and  $L$  is the width of the control limit, which in practice is usually specified as 3, which corresponding to a false alarm rate of 0.27 %. In this paper, we use a one-sided version of an EWMA for fall detection [40].

Accelerometer-based EWMA fall detection comprises two stages: training and testing. This approach is schematically illustrated in Fig. 4 and outlined in the following steps.

- (i) Training involves computation of a magnitude of acceleration from the fall-free data and the EWMA control limit for detection (see Eq. 3).
- (ii) Testing is a comparison of the magnitude of acceleration of the new measurements with the previously computed threshold for making decision.

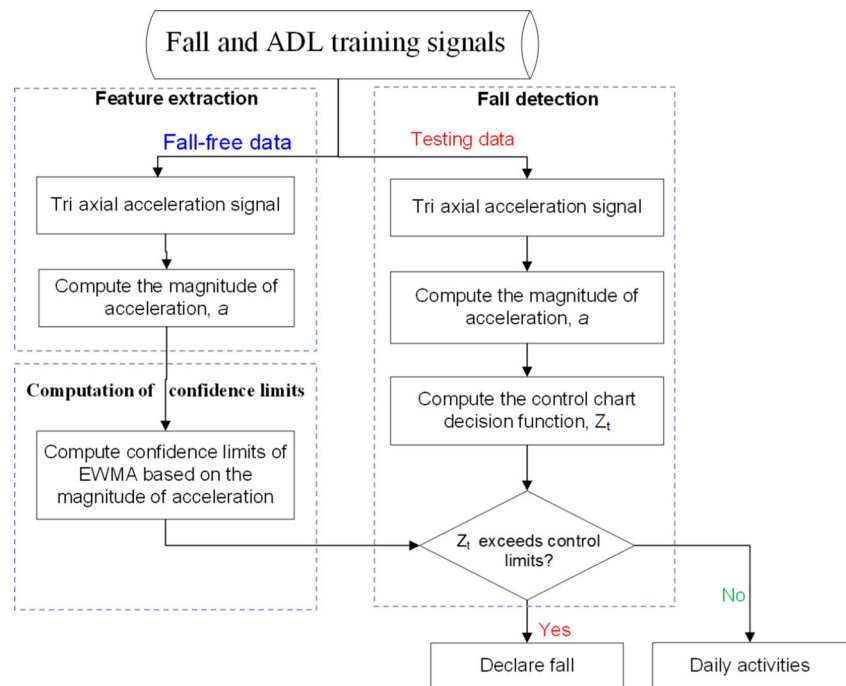
One disadvantage of anomaly detectors such as the EWMA chart, is that they can detect different types of anomalies, but they cannot distinguish the type of anomaly. In this paper, machine learning algorithms are applied to distinguish false falls from true falls. Specifically, the EWMA fall detector identifies a fall, the system extracts the monitored person and use the classifier to confirm the fall event. If the EWMA decision statistic is below the threshold, then new accelerometric data is collected. The classification stage is executed only when a potential fall is detected by EWMA scheme, which significantly reduces computational costs and false alarms.

### Classification strategy

During classification, only data from the camera are taken into account (i.e., no sensor data). In general, computer vision-based methods include four major steps:



**Fig. 4** The flow chart of the EWMA-based fall detection algorithm



data acquisition or collection, image segmentation, feature extraction, fall classification. During data acquisition, specific variables are recorded to determine whether the person has suffered a fall. In image segmentation, the body's silhouette is extracted from the image sequence and then used in the feature extraction step, which determines discriminative information needed as input in fall classification steps. The presence or absence of true falls is then distinguished by the fall classification phase where each sequence is diagnosed as fall or fall-like activities. In this section, we explain the extraction of the body's silhouette that is used to calculate the fall descriptors, discuss these descriptors and briefly describe the machine-learning techniques used to perform the classification task.

### Segmentation and preprocessing

For segmentation, the body's silhouette is extracted from the input image sequence using a background subtraction technique [49, 50]. The background image is defined as a

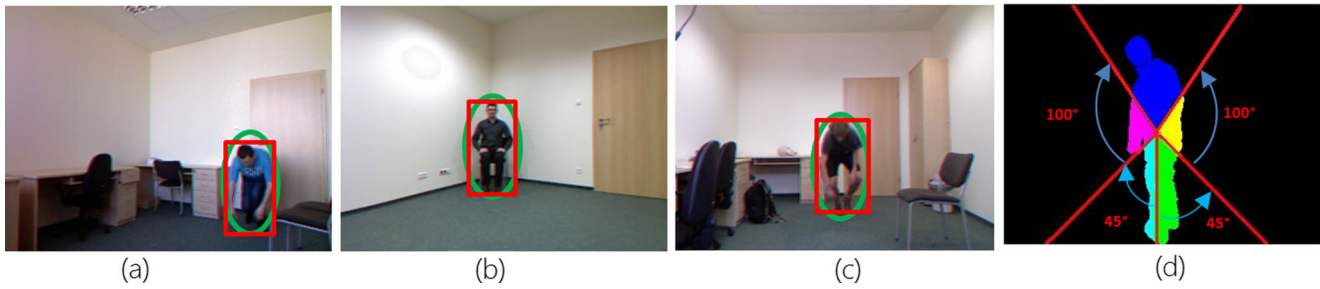
reference to eliminate the unchanged pixels in the frame sequences. This method is suitable here because it can manage multiple component models [50]. Figure 5 illustrates an example of the background subtraction algorithm, where background and input frames are represented by the two images on the left, respectively, and background subtraction frames before and after applying morphological operators, are shown on the right.

### Feature extraction

Accurate feature extraction is necessary for video-based fall classification, where extracted features have a direct impact on successful classification. The extracted features have to be invariant to image translation when the position of human body changes in the image and to scaling when the dimensions of the silhouette change as the distance between the monitored person and camera varies. Previous works have performed feature extraction by focusing on shape information to detect and classify falls; for example, the body's



**Fig. 5** Results of the background subtraction algorithm



**Fig. 6** Samples of partitioned human body areas

center of gravity [33], key joints [35], horizontal and vertical dimensions of the bounding box or approximated ellipse of the silhouette. However, these features cannot always distinguish among body postures, especially when there is a high degree of similarity between activities (e.g., dimensions and orientations of the ellipse are the same for both bending and sitting postures, as shown in Fig. 6b–c). For this reason, instead of using the body’s geometrical shape we base the extracted features on non-zero pixels that constitute the human body. More specifically, we use five partial occupancy areas (i.e., the number of pixels in each area of an image) of the body to detect and classify falls. These areas typically correspond to the action of body parts when in a standing posture. As shown in Fig. 6d, the body is divided into five portions. These areas were determined using the body’s center of gravity ( $x_G, y_G$ ), which is simply the barycenter of the pixels.

$$x_G = \frac{1}{N} \sum_{i=1}^N x_i, \quad y_G = \frac{1}{N} \sum_{i=1}^N y_i, \quad (4)$$

where  $N$  is the number of pixels representing the human body, and  $x_i$  and  $y_i$  denote the horizontal and vertical coordinates of the pixels composing the human body, respectively. Partitioning is performed by tracing five segments from the body’s center of gravity as shown in Fig. 6d. The first segment is vertical, the second and third segments are

located at  $45^\circ$  on either side of the vertical segment and the fourth and the fifth segments are situated at  $100^\circ$  on either side of the third and fourth segments (see Fig. 6d). The center of gravity and the five areas are computed for each image to represent the feature frame that we expect to be a precise characterization for human gesture classification.

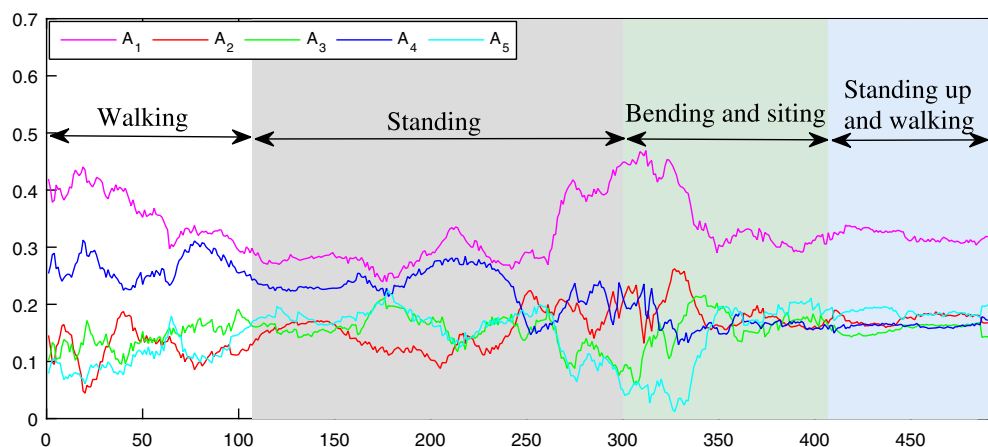
A normalization phase, where each area value is divided by the global body area, is performed to negate any concerns related to scaling. Given the total number of pixels making up the body area,  $A$ , and the number of pixels making up the partial areas  $A_i; i = 1 \dots 5$ , the normalized partial areas are given as:

$$R_i = \frac{A_i}{\sum_{i=1}^5 A_i}. \quad (5)$$

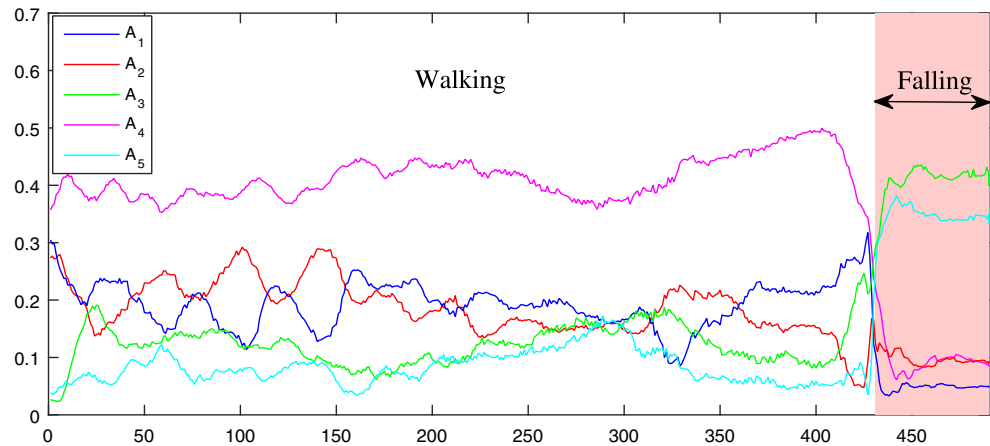
Figure 7 shows the time evolution of the five features ( $A_1 - A_5$ ) while daily activities are performed (walking, standing, bending and sitting, standing up) and Fig. 8 compares these features while walking to those while falling. We observe that these features change significantly just prior to and while falling, signaling the occurrence of an event that is significantly separable from the activities of daily living (ADL) (see Fig. 8).

Note that these ratios are invariant to translation and scaling, and that they take into account the rotation information necessary for fall detection. In other words, the extracted

**Fig. 7** Time evolution of features extracted from data during ADL (walking, standing, bending and sitting, and standing up)



**Fig. 8** Time evolution features extracted from data with fall between frame number 430 to 490



ratios are discriminative enough to describe human postures and computationally simple enough to allow for fast processing. Finally, since frames of a video sequence are assimilated to an observation sequence, the set of ratios that are computed for each frame are then concatenated to form the whole feature vector corresponding to the video sequence (see Fig. 9).

**Fall classification**

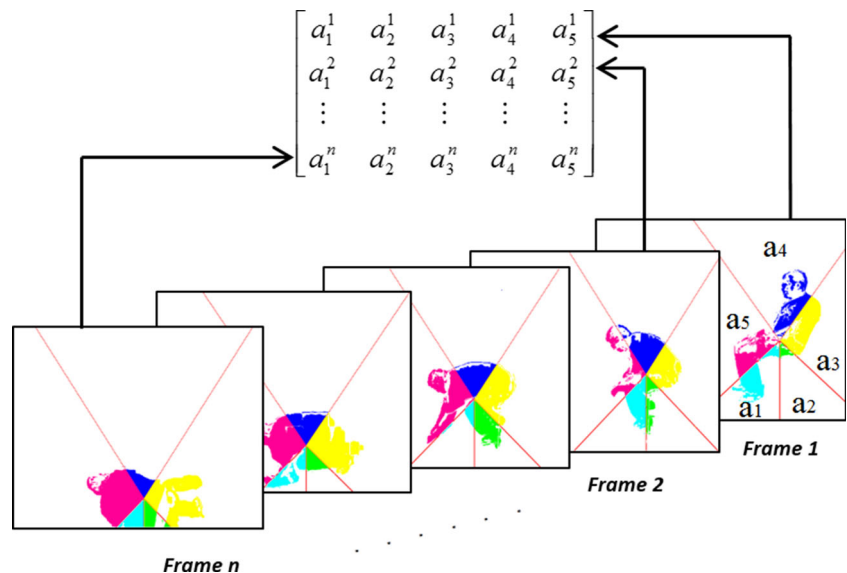
Now, we seek to discern between some intentional movements, considered to be fall-like activities, and real falls reported during the detection phase. Given a set of training data points with training labels, we now aim to determine the class label for an unlabeled set of testing data. Classification algorithms consist of training and testing phases. In training phase, a model is constructed from the training data (e.g., assign the system certain fall sequences as training samples) and the in the testing phase, the constructed model

is used to classify the remaining sequences (testing samples) according to their features. In this section, we provide a brief description of some most commonly used classifiers, namely SVM, neural network, naïve Bayes and K-Nearest Neighbor (KNN). In this work, we used only SVM for fall classification and the others are explored only for comparison. For more details about classification algorithms, please refer to [55].

*Support vector machine algorithm*

The SVM algorithm was first introduced by Vapnik [51], and it has since been used extensively in classification and detection [52]. The key idea behind the SVM algorithm is first to map the input space into a high-dimensional feature space (transformed space) via a kernel function, which makes the data approximately linear, and then to build an optimal separating hyperplane in the new space to distinguish between classes [51, 53]. The support vectors are

**Fig. 9** Features used to detect a fall





the training data points that are closest to the separating hyperplane; these points represent the maximum-margin hyperplane for the training data. In this way, SVMs look for a maximum margin hyperplane to separate data. Further details on SVM can be found in [51].

In this work, the SVM algorithm is used to classify falls. Although other algorithms may also have been suitable, the SVM classifier was selected for its ability to behave as both a linear and a nonlinear classifier through the exploitation of linear and nonlinear kernels, respectively. In this study, the SVM was employed to find a function (i.e., an optimal separating hyperplane) to map each detected fall into their corresponding labeled space  $y_k \in \{+1, -1\}$ , where +1 and -1 represent true fall and false fall, respectively.

The optimization separation function in SVM can be defined as follows:

$$\min_{w, \xi} \frac{1}{2} W^T W + C \sum_{i=1}^n \xi_i, \tag{6}$$

where  $C$  is the regularization parameter;  $\xi$  corresponds to slack variables introduced, which acts as a penalty term; and  $w$  represents the normal vector to the hyperplane. The optimal  $w$  is obtained as:

$$w = \sum_{i=1}^n y_i \alpha_i \varphi_i(x_i) \tag{7}$$

and the decision function is given by

$$f(x) = \text{sgn} \left( \sum_{i=1}^n y_i \alpha_i k(x_i, x_j) + b \right), \tag{8}$$

where  $\varphi_i(x_i)$  is the transformation applied to  $x_i$ ,  $\alpha_i$  corresponds to Lagrange multipliers,  $b$  is the bias term, and  $x_j$  is a test observation.  $K(x_i, x_j)$  represents the kernel function. Various kernel functions can be used for mapping processes [51].

For comparison sake, we tested neural network, K-nearest neighbor, and naïve Bayes classifiers. These classifiers were chosen because they are widely used in pattern recognition problems. A brief description of the three comparative classifiers is provided.

#### Neural network classification

Neural network classifier is a supervised algorithm that can use multiple input, output and hidden layers with an arbitrary number of neurons [54]. The most widely used neural classifier today is the multi-layer perception (network with the back propagation learning algorithm to optimize the network. The classic architecture of this classifier includes three layer types: input layers, hidden layers and output layers. Each layer comprises one or several neurons. The

input layers classify the features, the hidden layers are generally determined empirically and relatively to the expected classification accuracy and the output layer corresponds to the defined classes. Each class corresponds to a node in output layer. The output node value should provide the corresponding class for the input data (i.e., a high output value is expected on the correct class node and a low output value on all the rest). The optimal number of hidden layers and its corresponding neurons depends on each application. Although the neural network classifier has been applied to numerous fall detection applications [34], it is limited by using empirical risk minimization and non structural risk minimization and are more prone to over fitting problem.

#### K-nearest neighbor

The KNN algorithm is a non parametric classifier rested on the notion of a neighborhood [55, 56]. It is a simple classifier because there is no training involved and only distances between training samples are evaluated (e.g., the Euclidean distance function). The basic KNN algorithm uses the closest neighbors of the not-yet-classified new instances to classify them. Every time that a new example needs to be classified, it is compared with all the samples in the dataset. Consequently, KNN algorithms use a straightforward approach to solve classification problems [57].

#### Naïve Bayes classifier

Naïve Bayes classifier is a probabilistic supervised machine learning technique based on Bayes' theorem [58]. It assumes that all input data related to a class is independent of each other. During this stage, the algorithm learns the conditional probability of samples from the training data. During the test phase, the classification is accomplished by computing the posterior probability for all classes and then predicting the class with the highest probability value. This classifier is easy to implement and can achieve satisfactory results in most of the cases [59]. It handles missing values by ignoring the instance during probability estimate calculations; however, it is based on the independence assumption, which may not hold for some attributes and is often inappropriate for real world data.

#### Classification performance measures

Several measures can be used to evaluate the performance of a classifier, such as the receiver operating characteristic (ROC) analysis, the area under the curve (AUC) and the F-measure [60, 61]. The ROC curve is the simplest graphical tool for illustrating the performance of a classification algorithm. We recall that the ROC plots sensitivity, which is relative to the true positive rate (TPR) against 1-specificity,

which is relative to the false positive rate (FPR) over a range of the classification threshold.

$$\begin{aligned}
 TPR &= \frac{\text{Number of correctly classified falls}}{\text{Total number of fall incidents}}, \\
 FPR &= \frac{\text{Number of missclassified non-falls}}{\text{Total number of non-fall incidents}}.
 \end{aligned}
 \tag{9}$$

The AUC defines the quality of a classification method by using it as an index of accuracy, where an AUC of 1 corresponds to an ideal classification, an AUC of 0.5 corresponds to a chance classification, and an  $AUC > 0.9$  corresponds to a good classification. In this study, ROC curves were plotted and AUC values were then computed to compare classifications algorithms. We also evaluated the quality of classification methods using the F-measure. This coefficient examines the influence of random testing on the classification rate. It is computed by the combination of recall ( $r$ ) and precision ( $p$ ) ratios, where the recall represents the ratio of true positive number ( $tp$ ) to the total numbers of elements in the positive class, while the recall represents the ratio of the true positive number ( $tp$ ) to the sum true positive and false negative elements. The F-measure is expressed as follows [61]:

$$\begin{aligned}
 F &= (1 + \beta^2) \cdot \frac{p \cdot r}{(\beta^2 \cdot p) + r}, \text{ where } p = \frac{tp}{tp + fp} \text{ and} \\
 r &= \frac{tp}{tp + fn}.
 \end{aligned}
 \tag{10}$$

Because values of recall and precision ratios can vary from 0 to 1, the F-measure combines them into a single value for effectiveness. In this work the parameter  $\beta$  is fixed to 1.

### Experimental results

In this section, we evaluate the abilities of the EWMA fall detector with the SVM classifier to detect and classify a fall. The performance of this method is compared to that of conventional KNN, neural network, and naïve Bayes classifiers.

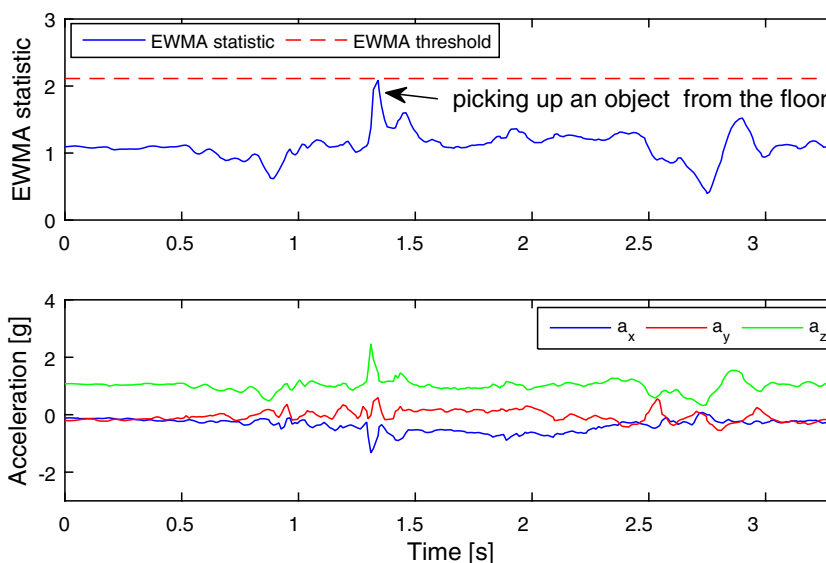
#### Detection results

This tests the effectiveness of the proposed EWMA fall detector through the practical accelerometer data obtained from the publicly available URFD datasets [10]. The URFD dataset comprises 70 (30 falls; 30 ADL, such as walking, sitting down, crouching down; and 10 sequences with fall-like activities, such as quickly lying on the floor and lying on the bed/couch) sequences with their corresponding accelerometric data recorded in different rooms (e.g., like office, classroom). All sequences containing falls include half from sitting on a chair and half from standing/walking. Fall events and ADL were measured from three volunteers over 26 years of age. Data from the three volunteers was recorded using using an x-IMU accelerometer, placed near the pelvis, with sampling rate of 256 Hz (see for details [10, 21]).

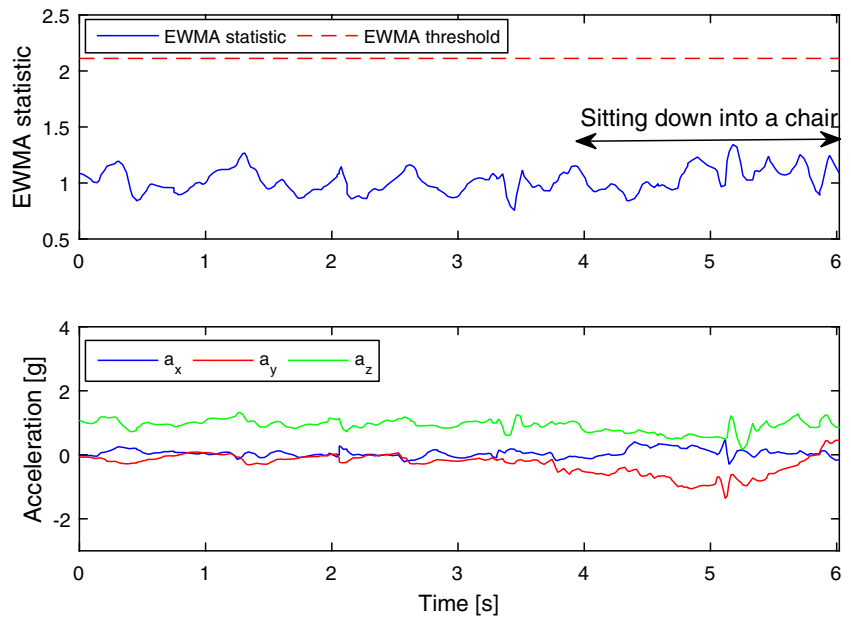
First, using training fall-free data, which contains ADL, such as picking-up an object from the floor, walking, crouching down, sitting down and standing up, the decision threshold of EMWA was computed based on the magnitude of acceleration. The EWMA scheme was constructed using the EWMA parameters  $\lambda = 0.4$  and  $L = 3$ , and the EWMA threshold value was found to be  $UCL = 2.12$ .

To show the effectiveness of the proposed strategy, three case studies are presented. In the first case some ADL, such

**Fig. 10** The time evolution of the EWMA statistic in the case of picking up an object from the floor (Case A, first example) (top) and its corresponding acceleration components (bottom). The horizontal dashed line denotes the control limit



**Fig. 11** The time evolution of the EWMA statistic in the case of sitting down into a chair (Case A, second example) (top) and its corresponding acceleration components (bottom). The horizontal dashed line denotes the control limit



as sitting down and standing up, picking up objects and sitting down on a chair are given (case A). In the second case, it is assumed that the testing data contains a true fall or a fall-like event (case B). In the third case, the testing data is assumed to contain false falls (i.e., fall-like activities, such as lying down) (Case C).

*Case A - monitoring fall-free ADLs activities*

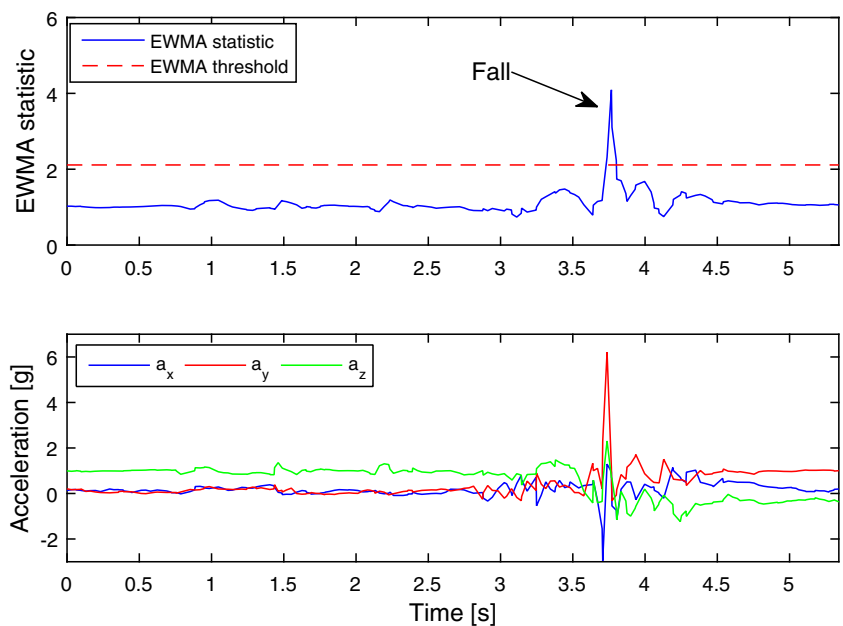
In the first case study, the EWMA monitoring chart is performed on fall-free data recorded from volunteers performing ADL. Two examples of daily activities, picking up an object from the floor and sitting down into a chair, are

presented. Figures 10 and 11 depict acceleration components of each sequence of the dataset and their corresponding EWMA decision results. A plot of the decision function of the EWMA chart (shown in Figs. 10 and 11) shows that the EWMA statistic is always below the threshold value and confirms that the data contains only normal daily activities, where no falls are present. The most pronounced EWMA value is at the instant of picking up an object from the floor.

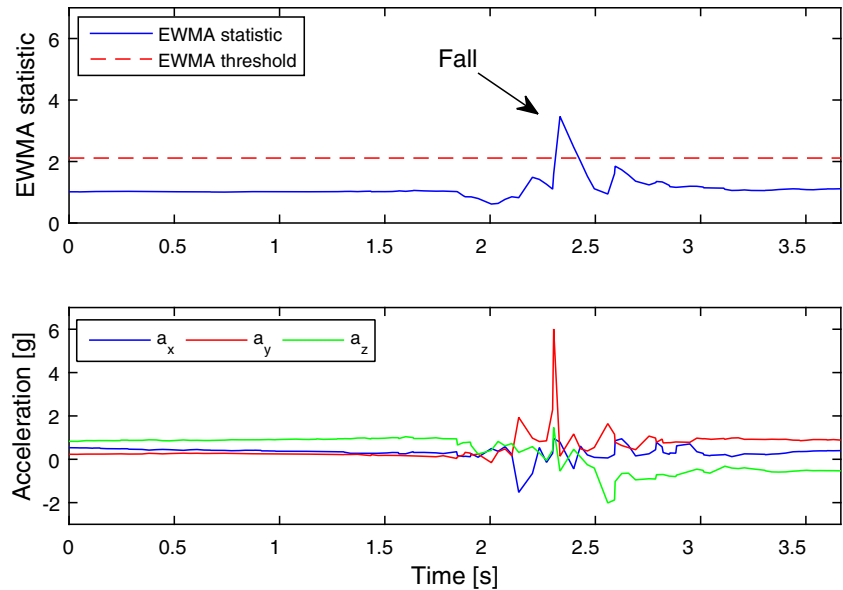
*Case B - true fall*

Two examples are presented here to illustrate the capacity of the EWMA fall detector to detect a true fall event. In

**Fig. 12** The time evolution of the EWMA statistic in the case of falling from a standing position (Case B, first example) (top) and its corresponding acceleration components (bottom)



**Fig. 13** The time evolution of the EWMA statistic in the presence of a fall event fall from a chair (Case B, second example). (top) and its corresponding acceleration components (bottom)



the first example, the testing data used to test the performance of the EWMA monitoring chart contain a fall from a walking or a standing position. The person stands at the beginning of a fall and then falls abruptly from standing to lying on the floor. The results of the EWMA chart and its corresponding accelerometric data are illustrated in Fig. 12, clearly showing that this chart detected the fall without false alarms.

A fall can occur not only when a person is standing, but also from a sitting position on a chair or from a lying on a bed during sleep. In the second example, the testing data contain a fall event from a chair. Monitoring results of the EWMA chart are shown in Fig. 13. The dashed red lines

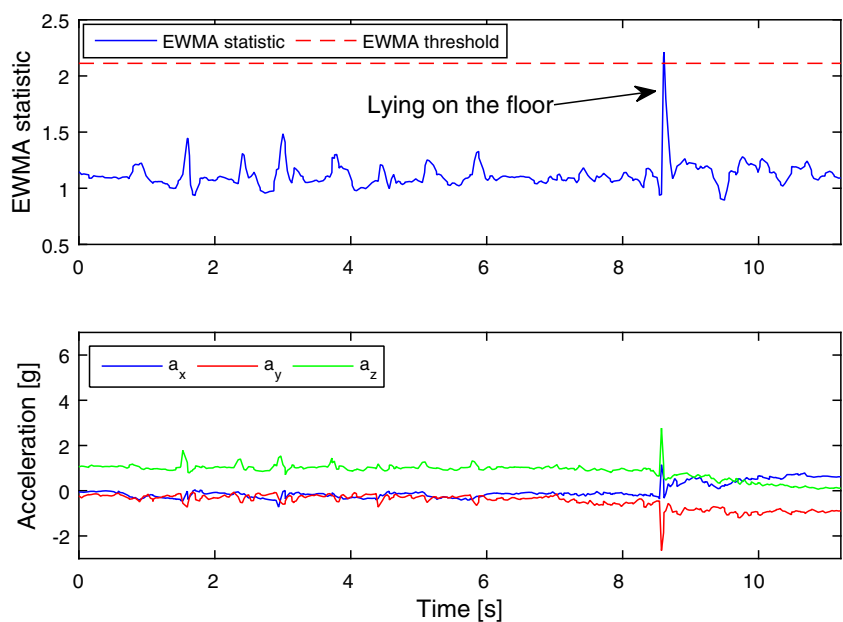
represent the 95 % confidence limit used to identify possible anomalies. The chart successfully detects this fall.

*Case C: False fall - Lying down*

In the third case study, the testing data contain lying down on the floor or false falls. The results using the EWMA chart (shown in Fig. 14) show that it could successfully detect false falls,

Because the EWMA chart cannot distinguish real falls from certain fall-like actions, such as lying down, a classification module should be added after the fall detection step.

**Fig. 14** The time evolution of the EWMA statistic in the presence of a fall event (Case C) and when  $\lambda = 0.4$ . The horizontal dashed line denotes the control limit



**Table 1** Performance comparison between EWMA-SVM, KNN, neural network and naïve Bays algorithms

Classifier	Accuracy	Se	Sp	Precession	Recall	F-measure	AUC
KNN	91.94	1	0.86	0.838	1	0.91	0.93
Neural network	95.16	1	0.91	0.903	1	0.95	0.95
Naive Bays	93.55	1	0.886	0.87	1	0.93	0.94
EWMA-SVM	96.77	1	0.9394	0.9355	1	0.9667	0.97

**Classification results**

To assess the detection ability of the classification strategy using the EWMA fall detector and SVM classifier (EWMA-SVM) fall classification strategy, we performed experiments on the publicly existing fall detection databases form URFD [10]. Falls and ADL used in this work are recorded with an RGB camera. The URFD dataset comprises 30 images per sequence for fall and typical ADL classes. All sequences are recorded with color cameras and synchronized with their corresponding accelerometer data. We evaluated the EWMA-SVM classifier and compared it with KNN, neural network, and naïve Bayes classifiers. We used a three-fold cross-validation to evaluate the classifiers.

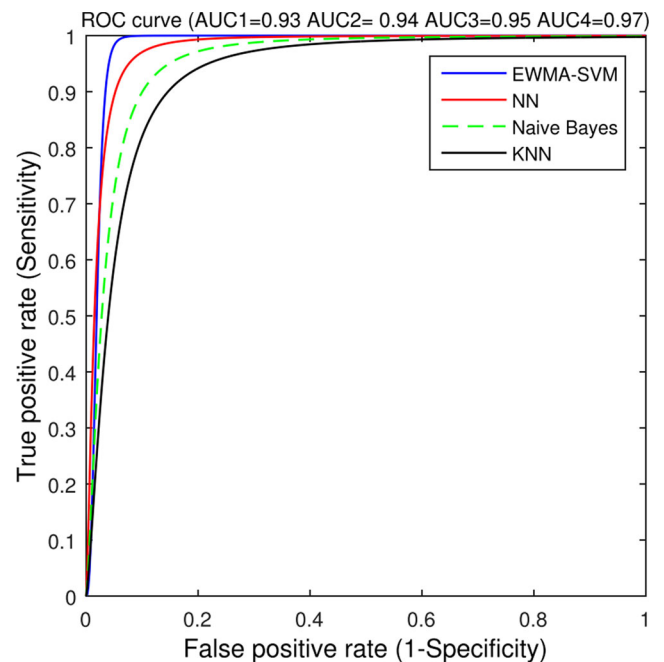
Table 1 compares the proposed fall detection and classification strategy of the EWMA-SVM with that using the other classifiers, KNN, neural network, and naïve Bayes, with detection phase. Note that because classification performance depends on the classifier’s parameters, parameters were optimized for each classification method during the training phase (parameter tuning phase), corresponding to maximum classification accuracy. For the neural network classifier, we varied the parameters to select the optimal multi layer perception architecture to achieve one hidden layer with fifteen neurons, which corresponded to the best classification rate. For the KNN classifier, the choice of  $k$  value depends strongly on the type of data. In this work, we varied the parameter  $k$  from 1 to 20 to establish that  $k = 3$  was the optimal value. For naïve Bayes classifier, a Gaussian model is adapted as predictor distribution model. The values of the corresponding mean and standard deviation are 3.43 and 0.38, respectively. Finally, we have iteratively tested the SVM-kernel parameters,  $\sigma$  the width of Gaussian kernel and  $C$  the parameter for the soft margin cost function, which controls the influence of each support vector;

**Table 2** Average training and testing processing times (s) for each classification method

Classifier	Training time (s)	Testing time (s)
KNN	0.22	
Neural network	2.65	0.15
Naive Bays	0.61	0.26
EWMA-SVM	0.182	0.0207

this process involves trading error penalty for stability. The pair with the highest accuracy was selected ( $\sigma = 0.125$  and  $C = 2^7 = 128$ ).

The results shown in Table 1 demonstrate that the integrated EWMA-SVM strategy is more accurate at detecting falls than any classifier alone, indicating that combining the detection phase with a classifier allows us to distinguish between daily activities and falls, reducing the space of training and testing data used as input for classification. The EWMA-SVM combination outperformed the neural network in the fall detection application likely because SVMs have a simple geometric interpretation and provide a sparse solution via structural risk minimization, unlike neural networks, which use empirical risk minimization. SVMs are less prone to over fitting than are neural networks. The EWMA-SVM outperformed KNN due to the ability of SVM to train important data sets, compared to the complexity of the KNN algorithm, which searches the nearest neighbors for each sequence sample. The EWMA-SVM combination also outperformed the naïve Bayes classification because of the independence assumption of the latter, which is far



**Fig. 15** Classification ROC curves of the EWMA-SVM, KNN, neural network and naïve Bays algorithms



**Table 3** Performance of fall detection on URFD data sequences

	Approach	Database	Overall Accuracy (%)
Kwolek et al. [21]	SVM	URFD (RGB, accelerometer and depth data)	94.28
	KNN	URFD (RGB, accelerometer and depth data)	95.71
The proposed approach	EWMA-SVM	URFD (RGB, accelerometer)	96.77

reaching and often inappropriate for a real world dataset. Furthermore, because the SVM classification is only applied to the data corresponding to detected cases, a reduced number of video sequences need to be classified, which makes processing much faster than when all data must be classified (see Table 2): where the whole set of video sequences is trained and classified into at least three classes (daily activities, intentional lying and real falls) to produce a complex training system that makes classifiers unsuitable for application to this type of data.

Table 2 presents the average processing times for each classification technique during training and testing phases. Processing time is a helpful tool for comparing the complexity of classifiers. During the learning phase, the neural network classifier requires the longest time to optimize its parameters, where EWMA-SVM presents the shortest processing time; however, during the testing phase, the neural network classifier has a lower processing time than does naïve Bayes or KNN classifiers. It important to note that the KNN classifier does not have an explicit training phase, thus, 0.22s can be considered as the processing time for both training and testing phases. Although this processing time is considered to be acceptable for this application, the EWMA-SVM combination remains significantly faster with a higher degree of accuracy than neural network, KNN, and naïve Bayes classifiers.

Figure 15 shows ROC curves corresponding to the proposed EWMA-SVM approach and to neural network, KNN, and naïve Bayes classifiers. The EWMA-SVM classification is shown to outperform neural network, KNN, and naïve Bayes classifiers ( $AUC_{KNN} = 0.93$ ;  $AUC_{NN} = 0.94$   $AUC_{NaiveBayes} = 0.95$  and  $AUC_{EWMA-SVM} = 0.97$ ).

To highlight the efficiency of the proposed approach in terms of recognition accuracy, a comparison with two well-known classifiers proposed in [21] is presented in Table 3. In [21], the authors applied both SVM and KNN classification to a set of camera sequences to obtain depth and accelerometric data. For comparison, experiments were conducted on the same publicly available URFD datasets, allowing for a fair evaluation. The results shown in Table 3 clearly highlight the advantage of the EWMA-SVM methods over detecting falls by the other two classifications (KNN and SVM). The reason EWMA-SVM fall detection method outperforms the KNN and SVM methods given in [21] lies in the fact that it uses the EWMA metric, which

is highly sensitive to anomaly. In [21], the detection phase is based on accelerometric data and the decision threshold is fixed manually, whereas the threshold in the EWMA is based on the weighted moving average of all available observations, a design that provides improved sensitivity to fall events. In addition, low computational cost make the EWMA chart easily implemented in real time. Furthermore, it appears that using pixel-based area ratios adequately describes movements made by the human body. Therefore, based on the results of these experiments, the EWMA-SVM approach provides a more accurate fall detection than previously existing methods.

## Conclusion

In this work, information from both acceleration sensors and cameras were used to design a reliable fall detection strategy. Unlike vision-based approaches, where a pretreatment step is necessary, acceleration sensors do not require a pretreatment phase. This speeds up the processing step, making it suitable for the detection phase; however because images from RGB camera provide more information, they are appropriate for the classification phase. The results show that the combined EWMA-SVM approach successfully separated true falls from false falls. The EWMA identified features corresponding to falls that are pertinent for fall classification tasks plays a key role in reducing the size of the features used as the input data for SVM classification, which significantly reduces the computational burden and achieves reasonable accuracy. Comparison of the EWMA-SVM approach with four other commonly classifiers demonstrated the superior classification capacity of EWMA-SVM.

Despite the promising results for fall detection and classification obtained using the EWMA-SVM strategy, the work carried out in this paper raises a number of question and provides some directions for future work. In particular, the following points merit consideration from researchers.

- Herein, the detection of the fall is done on the basis of accelerometric data and data from an RGB camera. In the dark, RGB cameras are not capable of extracting the human silhouette. Because falls often occur to elderly people who may be alone, we plan to investigate the

viability of using an infrared camera in the dark to capture the images of the human body needed for the camera detection stage.

- The URFD dataset uses fall events and ADL from volunteers, although over the age of 26, does not reflect the geriatric community. Moreover, it would be useful to incorporate more data inputs such as heart rate and blood pressure provided by a smartwatch or a smartphone to further enhance the effectiveness of a fall detection system.

**Acknowledgments** We would like to thank the reviewers of this article for their insightful comments, which helped us to greatly improve its quality. The authors (Nabil Zerrouki and Amrane Houacine) would like to thank the LCPTS laboratory, Faculty of Electronics and Informatics, University of Sciences and Technology HOUARI BOUMEDIENE (USTHB) for the continued support during the research. This publication is based upon work supported by King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No: OSR-2015-CRG4-2582.

## References

1. Ageing, W.H.O., and Unit, L.C., WHO global report on falls prevention in older age. World Health Organization, 2008.
2. Giannakouris, K., Ageing characterises the demographic perspectives of the European societies. *Stat. Focus* 72:2008, 2008.
3. Todd, C., and Skelton, D., What are the main risk factors for falls among older people and what are the most effective interventions to prevent these falls? Copenhagen, WHO Regional Office for Europe, 2004.
4. Yu, M., Miao, R., Adel, N., Mohsen, S., Wang, L., and Chambers, J., A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment. *IEEE Trans. Inf. Technol. Biomed.* 16(6):1274–1286, 2012.
5. Heinrich, S., Rapp, K., Rissmann, U., Becker, C., and König, H.-H., Cost of falls in old age: a systematic review. *Osteop. Int.* 21(6):891–902, 2010.
6. Soriano, T., DeCherrie, L., and Thomas, D., Falls in the community-dwelling older adult: a review for primary-care providers. *Clin. Inter. Aging* 2(4):545, 2007.
7. Delahoz, Y., and Labrador, M., Survey on fall detection and fall prevention using wearable and external sensors. *Sensors* 14(10):19806–19842, 2014.
8. Zweifel, P., Felder, S., and Meiers, M., Ageing of population and health care expenditure: a red herring? *Health Econ.* 8(6):485–496, 1999.
9. Mubashir, M., Shao, L., and Seed, L., A survey on fall detection: Principles and approaches. *Neurocomputing* 100:144–152, 2013.
10. Kwolek, B., and Kepski, M., Human fall detection on embedded platform using depth maps and wireless accelerometer. *Comput. Methods Programs Biomed.* 117(3):489–501, 2014.
11. Howcroft, J., Kofman, J., and Lemaire, E., Review of fall risk assessment in geriatric populations using inertial sensors. *J. Neuroeng. Rehab.* 10(1):1, 2013.
12. Rougier, C., Meunier, J., St-Arnaud, A., and Rousseau, J., Robust video surveillance for fall detection based on human shape deformation. *IEEE Trans. Circ. Syst. Vid. Technol.* 21(5):611–622, 2011.
13. Hazelhoff, L., and Han, J., et al.: Video-based fall detection in the home using principal component analysis. In: *Advanced Concepts for Intelligent Vision Systems*, pp. 298–309. Springer, 2008.
14. Vishwakarma, V., Mandal, C., and Sural, S., Automatic detection of human fall in video. In: *Pattern Recognition and Machine Intelligence*, pp. 616–623. Springer, 2007.
15. Liu, C.-L., Lee, C.-H., and Lin, P.-M., A fall detection system using k-nearest neighbor classifier. *Expert Syst. Appl.* 37(10):7174–7181, 2010.
16. Rimminen, H., Lindström, J., Linnavuo, M., and Sepponen, R., Detection of falls among the elderly by a floor sensor using the electric near field. *IEEE Trans. Inf. Techn. Biomed. Publ. IEEE Eng. Med. Biol. Soc.* 14(6):1475–1476, 2010.
17. Li, Y., Ho, K., and Popescu, M., A microphone array system for automatic fall detection. *IEEE Trans. Biomed. Eng.* 59(5):1291–1301, 2012.
18. Alwan, M., Rajendran, P., Kell, S., Mack, D., Dalal, S., Wolfe, M., and Felder, R., A smart and passive floor-vibration based fall detector for elderly. In: *Information and Communication Technologies, 2006. ICTTA'06. 2nd. Vol. 1*, pp. 1003–1007. IEEE, 2006.
19. Gibson, R., Amira, A., Ramzan, N., de-la Higuera, P.C., and Pervez, Z., Multiple comparator classifier framework for accelerometer-based fall detection and diagnostic. *Appl. Soft Comput.* 39:94–103, 2016.
20. Kwolek, B., and Kepski, M., Fuzzy inference-based fall detection using Kinect and body-worn accelerometer. *Appl. Soft Comput.* 40:305–318, 2016.
21. Kwolek, B., and Kepski, M., Improving fall detection by the use of depth sensor and accelerometer. *Neurocomputing* 168:637–645, 2015.
22. Tong, L., Song, Q., Ge, Y., and Liu, M., HMM-based human fall detection and prediction method using tri-axial accelerometer. *IEEE Sensors J.* 13(5):1849–1856, 2013.
23. Veltink, P., Bussmann, H., De-Vries, W., Martens, W., and Lummel, R., Detection of static and dynamic activities using uniaxial accelerometers. *IEEE Trans. Rehab. Eng.* 4(4):375–385, 1996.
24. Kangas, M., Konttila, A., Lindgren, P., Winblad, I., and Jämsä, T., Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait Posture* 28(2):285–291, 2008.
25. Purwar, A., Jeong, D., and Chung, W., Activity monitoring from real-time triaxial accelerometer data using sensor network. In: *International Conference on Control, Automation and Systems, ICCAS'07*, pp. 2402–2406. IEEE, 2007.
26. Igual, R., Medrano, C., and Plaza, I., Challenges, issues and trends in fall detection systems. *Biomed. Eng. Online* 12(66):1–66, 2013.
27. Bourke, A., O'donovan, K., and O'laighin, G., The identification of vertical velocity profiles using an inertial sensor to investigate pre-impact detection of falls. *Med. Eng. Phys.* 30(7):937–946, 2008.
28. Sposaro, F., and Tyson, G., ifall: an android application for fall monitoring and response. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2009*, pp. 6119–6122. IEEE, 2009.
29. Anderson, D., Keller, J., Skubic, M., Chen, X., and He, Z., Recognizing falls from silhouettes. In: *28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS'06*, pp. 6388–6391. IEEE, 2006.
30. Cucchiara, R., Prati, A., and Vezzani, R., A multi-camera vision system for fall detection and alarm generation. *Expert Syst.* 24(5):334–345, 2007.
31. Miaou, S.-G., Sung, P.-H., and Huang, C.-Y., A customized human fall detection system using omni-camera images and personal information. In: *1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare*, pp. 39–42. IEEE, 2006.
32. Jansen, B., and Deklerck, R., Context aware inactivity recognition for visual fall detection. In: *Pervasive Health Conference and Workshops*, pp. 1–4. IEEE, 2006.

33. Liu, H., and Zuo, C., An improved algorithm of automatic fall detection. *AASRI Procedia* 1:353–358, 2012.
34. Foroughi, H., Aski, B., and Pourreza, H., Intelligent video surveillance for monitoring fall detection of elderly in home environments. In: *11th International Conference on Computer and Information Technology, ICCIT 2008*, pp. 219–224. IEEE, 2008.
35. Bian, Z.-P., Hou, J., Chau, L.-P., and Thalmann, M.N., Fall detection based on body part tracking using a depth camera. *IEEE J. Biomed. Health Inf.* 19(2):430–439, 2015.
36. Stone, E., and Skubic, M., Fall detection in homes of older adults using the microsoft kinect. *IEEE J. Biomed. Health Inf.* 19(1):290–301, 2015.
37. Kadri, F., Harrou, F., Chaabane, S., Sun, Y., and Tahon, C., Seasonal ARMA-based SPC charts for anomaly detection: application to emergency department systems. *Neurocomputing* 173:2102–2114, 2016.
38. Cheng, W.-C., and Jhan, D.-M., Triaxial accelerometer-based fall detection method using a self-constructing cascade-adaboost-SVM classifier. *IEEE J. Biomed. Health Inf.* 17(2):411–419, 2013.
39. Cola, G., Avvenuti, M., Vecchio, A., Yang, G.-Z., and Lo, B., An on-node processing approach for anomaly detection in gait. *IEEE Sensors J.* 15(11):6640–6649, 2015.
40. Montgomery, D.C., *Introduction to Statistical Quality Control*. New York: Wiley, 2005.
41. Lucas, J., and Saccucci, M., Exponentially weighted moving average control schemes: properties and enhancements. *Technometrics* 32(1):1–12, 1990.
42. Roberts, S.W., Control chart tests based on geometric moving averages. *Technometrics* 1(3):239–250, 1959.
43. Harrou, F., Nounou, M., Nounou, H., and Madakyaru, M., PLS-based EWMA fault detection strategy for process monitoring. *J. Loss Prev. Process Ind.* 36:108–119, 2015.
44. Harrou, F., and Nounou, M., Monitoring linear antenna arrays using an exponentially weighted moving average-based fault detection scheme. *Syst. Sci. Control Eng. Open Access J.* 2(1):433–443, 2014.
45. Morton, A., Michael, W., Mary-Louise, M., Dobson, S., Looke, J., and Anna, S., The application of statistical process control charts to the detection and monitoring of hospital-acquired infections. *J. Qual. Clin. Pract.* 21(4):112–117, 2001.
46. Harrou, F., Nounou, M., and Nounou, H., A statistical fault detection strategy using PCA based EWMA control schemes. In: *9th Asian Control Conference (ASCC)*, pp. 1–4. IEEE, 2013.
47. Rabhu, S., and Runger, G., Designing a multivariate ewma control chart. *J. Qual. Technol.* 29(1):8–15, 1997.
48. Hunter, J.S., The exponentially weighted moving average. *J. Qual. Technol.* 18(4):203–210, 1986.
49. Elgammal, A., Harwood, D., and Davis, L., Non-parametric model for background subtraction. In: *Computer Vision/ECCV 2000*, pp. 751–767. Springer, 2000.
50. Kim, K., Chalidabhongse, T., Harwood, D., and Davis, L., Real-time foreground-background segmentation using codebook model. *Real-Time Imag.* 11(3):172–185, 2005.
51. Vapnik, V., *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2013.
52. Yin, Z., and Hou, J., Recent advances on SVM based fault diagnosis and process monitoring in complicated industrial processes. *Neurocomputing* 174:643–650, 2016.
53. Hsu, C.-W., and Lin, C.-J., A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* 13(2):415–425, 2002.
54. Hertz, J., Krogh, A., and Palmer, R., *Introduction to the Theory of Neural Computation*. Vol. 1, 1991.
55. Duda, R., Hart, P., and Stork, D., *Pattern Classification*: Wiley, 2012.
56. Cover, T., and Hart, P., Nearest neighbor pattern classification. *IEEE Trans. Inf. theory* 13(1):21–27, 1967.
57. Kepski, M., and Kwolek, B., Fall detection using ceiling-mounted 3d depth camera. In: *International Conference on Computer Vision Theory and Applications (VISAPP)*. Vol. 2, pp. 640–647. IEEE, 2014.
58. Jordan, A., On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. *Adv. Neural Inf. Process. Syst.* 14:841, 2002.
59. Zhang, H., and Su, J., Naive bayes for optimal ranking. *J. Exper. Theor. Artif. Intell.* 20(2):79–93, 2008.
60. Wang, W., Chen, S., and Qu, G., Incident detection algorithm based on partial least squares regression. *Transp. Res. Part C: Emerg. Technol.* 16(1):54–70, 2008.
61. Zerrouki, N., and Houacine, A., Automatic classification of human body postures based on curvelet transform. In: *Image Analysis and Recognition*, pp. 329–337. Springer, 2014.