Design of Elderly Fall Detection Based on XGBoost

Min Xiao[1(✉)], Yuanjian Huang[1], Yuanheng Wang[1], and Weidong Gao[2]

[1] School of Computer Science and Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
xiaomincloud@163.com, {huangyj, jahracal}@bupt.edu.cn
[2] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
gaoweidong@bupt.edu.cn

Abstract. Aging has become a serious problem facing the whole world. Falling is the leading cause of injury and death in the elderly. This paper proposes a fall detection algorithm based on machine learning XGBoost and full-field positioning. Using the data of gyroscope and acceleration sensor, we exploit the "full-field positioning" to increase the dimension of input data and propose a method "maximum satisfaction rate" to mark and train the threshold of data. The experimental results show that this design has obtained high accuracy on falling detection and perfect balance between sensitivity and specificity.

Keywords: Fall detection · XGBoost machine learning · Gyroscope · Full-field positioning

1 Introduction

The WHO has pointed out that at present world's aging trend is becoming more and more serious. Until 2017, the elderly aged 65 and over accounted for 8.696% of the world's population [1]. The physiology of the human body is degraded as we grow older. And coupled with the complexity of the living environment, the elderly are prone to accidents such as falls. Moreover, some elderly people are alone at home, and no one is around when they fall, which causes serious consequences. The 2018 World Health Organization showed that falls are the second leading cause of accidental or unintentional injury deaths worldwide. Besides, adults older than 65 years of age suffer the greatest number of fatal falls [2]. Thus, a family using automatic device is in urgent need to protect the elderly from falling injury.

In recent years, there have been many research results of fall detection in the elderly at home and abroad: wearable fall recognition alarm system, fall alarm system based on embedded vision, elderly fall detection based on STM32 system, fall detection belt based on accelerometer, wearable fall monitoring, and so on. All fall detection products can be divided into two categories: visual inspection-based methods and sensor-based methods.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 46–57, 2020.
https://doi.org/10.1007/978-981-15-0187-6_6

(1) Vision-based methods: such as “multi-camera video surveillance system” [3]. First of all, they have great limitations and cannot be popularized in every household, which can only be applied to specific occasions. Secondly, the visual product judges whether the fall is theoretically based on the appearance of the human body falling, and the detection accuracy is difficult to be improved in essence.
(2) Sensor-based methods have fewer features to judge first, which are generally based on acceleration or angular velocity, or both. Such as the “wearable sensors for reliable fall detection,” it just judges the falling by the threshold of acceleration [4]. And “elderly fall monitoring method and device” [5] pointed out that using acceleration and angle to monitor falling, and “fall detection analysis with wearable MEMS-based sensors” [6] has consistent ideology with our paper to some degree. However, it just proposed a method and the execution of method is unknown.

Based on the survey of the above products or papers: If we want to put such products into wide application, there are two main problems to be solved: The first is the high accuracy with the balance between sensitivity and specificity, and the second is household, which means simplicity and convenience. Considering that XGBoost has higher accuracy and lower false-positive rate than other algorithm, it is a kind of gradient boosting which has proven many times to be an effective prediction algorithm for both classification and regression task, such as crude oil price forecasting [7], DDoS attack detection [8], and so on. Thus, we decide to use XGBoost as the basis of detection.

2 Proposed Falling Detection Based on XGBoost

2.1 The Process of Establishing the Model

The falling process includes:

(1) The horizontal acceleration relative to the heaven and earth coordinate system increases: This phenomenon occurs because during the movement of the human body, the horizontal acceleration increases and the speed decreases due to being tripped.
(2) The *X*-axis angular velocity and *Y*-axis angular velocity relative to the heaven and earth coordinate system increase.
(3) The vertical acceleration relative to the heaven and earth coordinate system increases: This is caused by the interaction between human body and the ground.
(4) The static process after falling, during which the human body’s pitch angle will reduce (that is, the included angle between the human body and the horizontal ground becomes smaller).

Generally, unless the person faints after falling, it is difficult to detect this very weak stationary process. Process (2) is a necessary condition for the fall to occur; that is, it is considered to have fallen when the body is dumped. The occurrence of the process (3) substantively causes harm to the human body, and the alarming of damage is the purpose of detecting the fall. The data we collected using gyroscopes and accelerometers is shown in Fig. 1. Process (4) also has the same effect when “lying down,” so the angle is introduced as an auxiliary judgment condition.

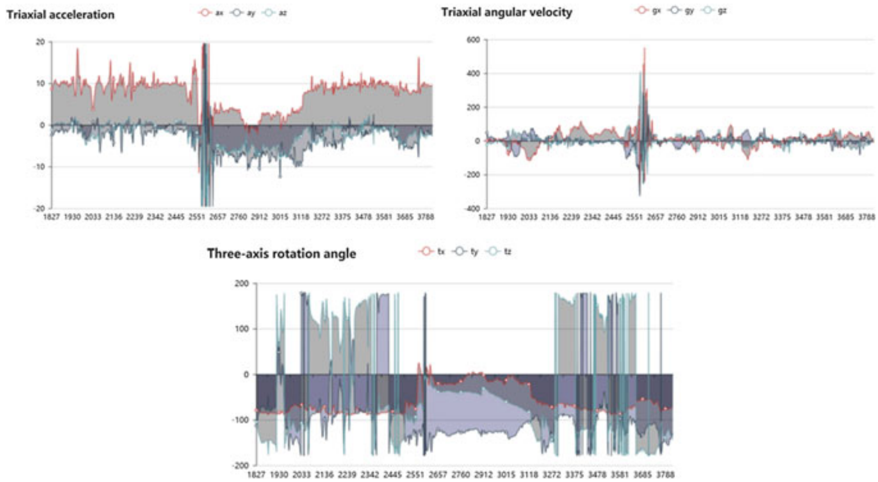

Fig. 1. Triaxial acceleration, triaxial angular velocity, triaxial rotation during the falling

The triaxial acceleration, triaxial angular velocity, triaxial rotation angle during the fall are shown in Fig. 1 (the abscissa unit is 10 ms, and the ordinate unit is m/s^2).

To detect whether a fall has occurred, the core is to distinguish between daily life behavior and fall behavior. Thus, we collected a large amount of data from different scenes in daily life, including running, squatting, going upstairs and downstairs, walking, lying down, and jumping.

Based on the various types of data collected above, we first draw some brief conclusions:

(1) The essence of falling damage to the human body is “force.” According to Newton’s first law, the greater the acceleration, the greater the external force the human body receives, and the greater the damage. Then, the combined acceleration “azz” is used as one of the judgment conditions for determining the occurrence of the falling.

$$\mathrm{azz} = \sqrt{a_x^2 + a_y^2 + a_y^2} \tag{1}$$

(2) The falling causes the human body to change from an “upright” state to a “squat” state. During this process, the angular velocity in the horizontal direction changes. Thus, the change of angular velocity is measured by the horizontal angular velocity “gyro.” The horizontal angular velocity is [4]:

$$\text{gyro} = \sqrt{g_x^2 + g_y^2} \tag{2}$$

(3) During the process of collecting data, we found that in some cases, the occurrence of acceleration peaks has a certain degree delay compared to the angular velocity peaks. The result of the analysis is the same: When the human body falls, the upper body is first dumped. In this process, the angular velocity reaches a peak at a certain moment; when the human body touches the ground, the acceleration reaches a peak. In addition, the acceleration can also reach an acceleration close to the fall when running and jumping; thus, we consider the importance of angular velocity > the importance of acceleration.

(4) According to the data analysis (Table 1), in order to distinguish the acceleration and angular velocity between falling and daily activities, the angle is introduced to judge the falling. Due to the "90° Euler angle limitation" and the complicated triangulation in real time, we choose to use the quaternion method to calculate the angle. Quaternion: An object can reach any attitude by rotating around an axis to a certain angle. In the process of calculating the angle, the "full-field positioning" algorithm is used combined with the real-time acceleration correction on angular velocity: According to the previous gravity unit vector and the gravity unit vector obtained by this measurement, the cross-product of two is obtained for correcting angular velocity. Then, quaternion (q_0, q_1, q_2, q_3) is updated according to "Runge–Kutta of the first order," so that the triaxial rotation angles "t_x", "t_y", and "t_z" relative to the heaven and earth coordinate system are [9, 10]:

Table 1. Three-dimensional data in different scenes

Activities	Maximum combined acceleration (set)	Minimum combined acceleration (set)	Maximum XOY plane angular velocity (set)	Maximum pitch angel (set)
Running	22.084601	0.854584	133.284490	8.411562
Squatting	18.697620	4.635264	94.091349	71.766807
Walking	20.132368	3.359908	106.089583	63.725275
Up and down stairs	15.677989	4.344686	93.901372	62.253342
Lie down	12.272055	8.361481	155.291657	0.380015
Jumping	20.124672	1.625603	98.638090	23.801800
Falling	28.624261	4.920240	272.268783	17.015853

$$t_x = a\sin(-2 * q_1 * q_3 + 2 * q_0 * q_2) * 57.3 \tag{3}$$

$$t_y = a\tan 2(2 * q_2 * q_3 + 2 * q_0 * q_1, -2 * q_1 * q_1 - 2 * q_2 * q_2 + 1) * 57.3 \tag{4}$$

$$tz = a\tan 2(2 * q_1 q_2 + 2 * q_0 q_3, -2 * q_2 q_2 - 2 * q_3 q_3 + 1) * 57.3 \tag{5}$$

(5) In the same way as (2), we hope the angle changes horizontally; thus, we use the human pitch angle to judge the falling. The *X*-axis rotation angle is "t_x", and the *Y*-axis rotation angle is "t_y", whereby the XOY plane rotation angle "agz" is:

$$\mathrm{agz} = \arctan\left(\frac{1}{\sqrt{\tan^2 t_x + \tan^2 t_y}}\right) \tag{6}$$

2.2 Notation

Notation	Meaning
a_x	*X*-axis acceleration
a_y	*Y*-axis acceleration
a_z	*Z*-axis acceleration
g_x	*X*-axis angular velocity
g_y	*Y*-axis angular velocity
g_z	*Y*-axis angular velocity
gyro	XOY plane angular velocity
t_x	*X*-axis rotation angle
t_y	*Y*-axis rotation angle
t_z	*Z*-axis rotation angle
agz	Pitch angle
Win	Detection window size
T	Sampling frequency
$\mathrm{Meet}_{\mathrm{azz}}[t]$	The number of "azz" meets the threshold in [*t*-300, *t*]
$\mathrm{Meet}_{\mathrm{gyro}}[t]$	The number of "gyro" meets the threshold in [*t*-300, *t*]
$\mathrm{Meet}_{\mathrm{agz}}[t]$	The number of "agz" meets the threshold in [*t*-300, *t*]
BORDER_AZZ	The threshold of "meet(azz)" in window
BORDER_GYRO	The threshold of "meet(gyro)" in window
BORDER_AGZ	The threshold of "meet(agz)" in window
q_0, q_1, q_2, q_3	Real-time quaternion

2.3 Falling Detection Model's Establishment

(1) "Sliding window": This paper proposes the "sliding window processing data" method to store updated data. The data storage includes whether the updated data satisfies the condition of the fall and the overall data of the window after updating the data. The single data is judged according to the XGBoost training model. We set the data judgment window size to "Win"; that is, we use the "Win" group data collected from a certain moment to present to determine whether a fall has occurred at present. Use a variable "forsample" to save the location of the update at present, and slide to the next position after the update. At the same time, an array of "Result_all [5]" is used to save the number of results which is judged to be "True."

(2) "Two sets of falling detection model": A simple analysis of the data in "2. Model Establishment Process" shows that the angular velocity is the physical quantity that best describes the falling and the most accurate judgment of falling. Therefore, we believe that angular velocity reaching the threshold is first when falling occurs.

During the actual test, it was found that the acceleration of the sliding fall was greater than the acceleration of the forward fall. We theoretically analyze: Most of the subjects subconsciously supported the ground in the forward fall, which alleviates the impact. Thus, the acceleration was not big. Therefore, we put forward the idea of "grouping judgment": Under the premise of angular velocity characteristics, two sets of acceleration judgments are made: One is when the pitch angle reaches the threshold, and the acceleration reaches a relatively small threshold, it will be considered to have a side fall or a back fall; the other is when the acceleration reaches a relatively large threshold, it is considered to have a forward fall. And the "grouping judgment" optimizes the judgment model.

In addition, we can use two sets of falling detection models to balance sensitivity and specificity [11]. Specificity and sensitivity are defined as follows:

$$\text{Sens} = \frac{\text{True falling}}{\text{True falling} + \text{False falling}} \tag{7}$$

$$\text{Spec} = \frac{\text{non-falling}}{\text{non-falling} + \text{False non-falling}} \tag{8}$$

Assuming: For n sets of data, the actual falling data accounted for "f"; the accuracy of classifier 1 to falling and non-falling data is (p_{11}, p_{12}); the accuracy of classifier 2 to falling and non-falling data is (p_{21}, p_{22}); assume that under this model, the applying frequency weight of the two models is w_1, w_2.

When there only exists classification 1:

$$\text{Sens} = \frac{n \cdot f \cdot p_{11}}{n \cdot f \cdot p_{11} + n \cdot (1-f) \cdot (1-p_{12})} \tag{9}$$

$$\text{Spec} = \frac{n \cdot (1-f) \cdot p_{12}}{n \cdot (1-f) \cdot p_{12} + n \cdot f \cdot (1-p_{11})} \tag{10}$$

After introducing classification 2, the overall specificity and sensitivity can be balanced by adjusting the accuracy of classification 2:

$$\text{Sens}' = \frac{n \cdot f \cdot \frac{w_1 \cdot p_{11} + w_2 \cdot p_{21}}{w_1 + w_2}}{n \cdot f \cdot \frac{w_1 \cdot p_{11} + w_2 \cdot p_{21}}{w_1 + w_2} + n \cdot (1-f) \cdot \left(1 - \frac{w_1 \cdot p_{12} + w_2 \cdot p_{22}}{w_1 + w_2}\right)} \tag{11}$$

$$\text{Spec}' = \frac{n \cdot (1-f) \cdot \frac{w_1 \cdot p_{12} + w_2 \cdot p_{22}}{w_1 + w_2}}{n \cdot f \cdot \frac{w_1 \cdot p_{12} + w_2 \cdot p_{22}}{w_1 + w_2} + n \cdot (1-f) \cdot \left(1 - \frac{w_1 \cdot p_{11} + w_2 \cdot p_{21}}{w_1 + w_2}\right)} \tag{12}$$

① If $p_{21} > p_{11}$, that is comparing to classifier 1, classifier increases the accuracy for falling data. Then, the corresponding accuracy for non-falling data will be reduced: $p_{22} < p_{12}$. According to (11)(12), Sens' ↓ and Spec ↑.

② If $p_{21} > p_{11}$, that is comparing to classifier 1, classifier reduces the accuracy for falling data. Then, the corresponding accuracy for non-falling data will be increased: $p_{22} < p_{12}$. According to (11)(12), Sens' ↑ and Spec ↓.

2.4 Data Threshold Training Based on XGBoost

According to the data analysis results, and combined with the actual situation, we use 3 s data (setting T = 100 Hz, then Win = 300, the corresponding window data size is 300 groups) as a critical window to determine whether a person has fallen; that is, the feature value of the fall occurred within the last 3 s. Before training for XGBoost, we first need to manually mark our falling data. In order to make the training results more accurate, we make preset values for the threshold.

According to the data collected by daily behavior, the three-dimensional data in different scenarios is shown in Table 2.

Table 2. Three-dimensional data preset

The angular velocity threshold of single group	gyro > 133.2844 deg/s
The acceleration threshold of single group	azz < 3.35 m/s^2 or azz > 12.2720 m/s^2
The pitch angle threshold of single group	agz < 38.3898°

The data characteristics of the normal fall are: The combined acceleration increases, the XOY plane angle velocity increases, and the human pitch angle becomes smaller. As shown above, the red color marked data is the closest to the fall feature, and the yellow color marked data is relatively closer to the fall feature. It can be observed that the running data characteristics are the closest to falling. We expected that it will be better for threshold to differentiate the falling and daily activities. At the same time, considering the sensitivity of the fall detection comprehensively, we preset the three-dimensional data threshold as shown in Table 2.

In this project, the difficulty of data training is how to choose the feature area where the fall occurs. Therefore, we propose a "maximum satisfaction rate" method: The number of single-group data satisfying the above threshold conditions within a certain 3 s: azz < AC2 OR azz > AC1, gyro > AG, agz < AN, is represented as meet_azz[t], meet_g[t], meet_agz[t]; then:

$$\text{meet}_{\text{azz}}[t] = \sum_{t-300}^{t} \text{int } (\text{azz}[t]\langle AC2\text{azz}[t]\rangle AC1) \tag{13}$$

$$\text{meet}_{\text{gyro}}[t] = \sum_{t-300}^{t} \text{int}(\text{gyro}[t] > AG) \tag{14}$$

$$\mathrm{meet}_{\mathrm{agz}}[t] = \sum_{t-300}^{t} \mathrm{int}(\mathrm{agz}[t] < AN) \tag{15}$$

For a fall process, we think that when meet_azz[t], meet_gyro[t], and meet_agz[t] all reach the maximum value, the presets are adjusted according to the size of each "satisfaction rate" meet[*t*] in different scenarios. At last, the falling is considered to occur in a time interval [*t*-300, *t*]. Thus, among these 300 sets of data, only the data satisfying the above threshold condition is marked.

After the marking is completed, we put all three-dimensional data into XGBoost for training. At the same time, we introduce the following three concepts: accuracy, recall, and f1_score to measure the training effect. TP is the number of falling data (positive class), TN is the number of non-falling data (negative class), FP is the number of negative class divided into positive class, and FN is the number of positive class divided into negative class. Thus, we define [12]:

$$\mathrm{accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{16}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{17}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{18}$$

$$\frac{2}{\mathrm{F1\_score}} = \frac{1}{\mathrm{Precision}} + \frac{1}{\mathrm{Recall}} \tag{19}$$

The 25,000 sets of training data contain 10 times falling marked data. Besides, it does not include daily activities' data, the reason is that for single-group data, the falling data accounts for a little part, and the daily activities' data value is usually close to that of the falling. Thus, the core of difference will be laid in the value of the "satisfaction rate." First, the training result of single set of data is as follows [13, 14]:

According to the decision tree obtained by XGBoost training, the simplified results are shown in Table 3 (Fig. 2):

For the above-mentioned preprocessed single-group data threshold, the number of data sets satisfying the above threshold in 3 s is collected in the same way. The maximum values in different scenarios are as follows (Table 4):

Similarly, based on the preset threshold, the adjustment is: BORDER_AZZ = 30, BORDER_GYRO = 8, BORDER_AGZ = 25. The values satisfy 90.90, 72.72, 88.89% of the database fall data.

Table 3. XGBoost three-dimensional threshold training results

Single data angular velocity threshold	gyro > 13 3.6826 deg/s
Single data acceleration threshold 1	azz < 3,9453 m/s^2 or azz > 13.3214 m/s^2
Single data pitch angle threshold	agz < 19.9387° or 29.37185° < agz < 38.3750°

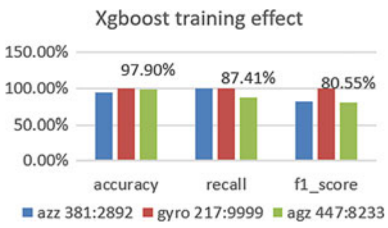

Fig. 2. XGBoost training effect

Table 4. Maximum "window satisfaction rate" in different scenarios

Activities	Maximum "window satisfaction rate" of combined acceleration (set)	Maximum "window satisfaction rate" of XOY plane angular velocity (set)	Maximum "window satisfaction rate" of pitch angle (set)
Running	205	3	40
Squatting	61	0	0
Walking	94	1	0
Upstairs and downstairs	62	0	0
Lie down	0	1	300
Jumping	92	0	22
The improved average "satisfaction rate" when falling	35	24	38

3 Performance Evaluation

For the threshold of acceleration 1, a person weighted 60 kg will receive a force of 770 N at the peak of the acceleration. The pressure applied by falling is the same as a free fall egg from 5th floor. It can be seen that the threshold is reasonable, and under this force, human body is indeed injured. And it is same as threshold 2 (Table 5).

The overall fall accuracy is relatively high, and the sensitivity and specificity are relatively balanced; thus, the detection effect is excellent. According to the data

Table 5. Parameter of the flowchart

Single set of data angular velocity threshold	gyro > 272.2687 deg/s
Single set of data acceleration velocity threshold 1	azz < 3.9453 m/s^2 or azz > 1 2.8214 m/s^2
Single sat of data acceleration velocity threshold 2	azz < 3.9453 m/s^2 or azz > 15.3214 m/s^2
Single set of data angle threshold	agz < 19.9387° or 29.37185° < agz < 38.3750°
The threshold of "window satisfaction rate": BORDER AZZ	30
The threshold of "window satisfaction rate": BORDER. GYRO	8
The threshold of "window satisfaction rate": BORDER AGZ	25
The accuracy of fall detection (based on 91 sets of data, which includes 59 sets of non-falling data and 32 sets of tailing data)	90.11%
Sensitivity	89.66%
Specificity	90.32%

analysis, the reason for the misjudgment to fall data is that the tester's action is too intense, making the action close to falling, thereby alarming. However, the false alarm with a small proportion can be ignored because it will not cause harmful consequences. For the case of falling but no alarm, we analyze: Some falls belong to "pseudo-fall"; that is, all the required features have occurred, but the data has not reached the threshold, and although it will not cause harmful consequences theoretically, the accuracy of this aspect needs to be improved.

The algorithm flowchart is shown in Fig. 3.

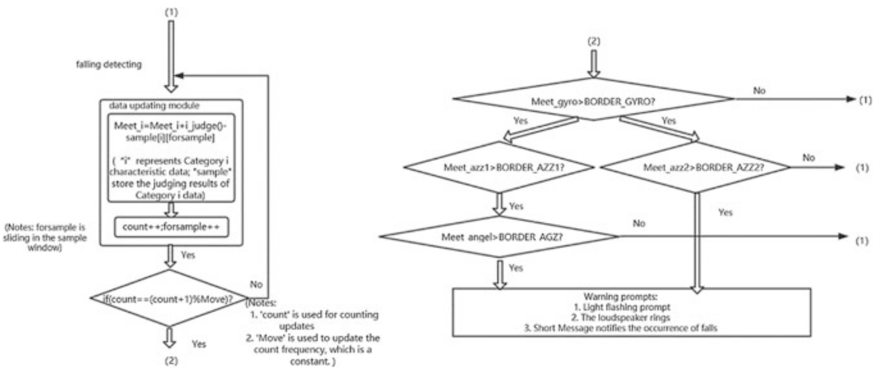

Fig. 3. Flowchart of falling algorithm

4 Conclusions

In this paper, we study fall detection algorithm based on machine learning. We extract the combined acceleration, the angular velocity of XOY plane, and the pitch angle as the characteristics of the falling process, in which pitch angle is introduced to distinguish the two sets of falls. In the preprocessing of data, we exploit the "full-field positioning" to obtain the pitch angle. Then, we proposed "sliding window method" to update the data. To solve the difficulty in locating the occurrence of falling, we proposed the "maximum satisfaction rate" method, which affects the marking of falling process in turn. Then, the XGBoost is used for threshold accurately training. Finally, the falling detection model we designed has received high accuracy with perfect balance between the sensitivity and specificity.

Acknowledgements. This work is supported by the 2016 Computer and Science Institute, Beijing University of Posts and Telecommunications.

This work is also supported by National Key R&D Program of China under grant number SQ2018YFC200148-03.

References

1. "Falls". World Health Organization. 16 January 2018. https://www.who.int/news-room/fact-sheets/detail/falls
2. The World Bank (2017) Population ages 65 and above (% of total). https://data.worldbank.org/indicator/SP.POP.65UP.TO.ZS
3. Shieh W-Y, Huang J-C (2009) Speedup the multi-camera video-surveillance system for elder falling detection. In: 2009 International conference on embedded software and systems. IEEE
4. Chen J et al (2006) Wearable sensors for reliable fall detection. In: 2005 IEEE Engineering in medicine and biology 27th annual conference. IEEE
5. Petelenz TJ, Peterson SC, Jacobsen SC (2002) Elderly fall monitoring method and device. U. S. Patent No. 6,433,690. 13 Aug 2002
6. Yuan X et al (2015) Fall detection analysis with wearable MEMS-based sensors. In: 2015 16th International conference on electronic packaging technology (ICEPT). IEEE
7. Gumus M, Kiran MS (2017) Crude oil price forecasting using XGBoost. In: 2017 International conference on computer science and engineering (UBMK). IEEE
8. Chen Z et al (2018) XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud. In: 2018 IEEE international conference on big data and smart computing (BigComp). IEEE
9. Diebel J (2006) Representing attitude: euler angles, unit quaternions, and rotation vectors. Matrix 58(15-16):1–35
10. Shoemake K (1985) Animating rotation with quaternion curves. In: ACM SIGGRAPH computer graphics, vol 19, no 3. ACM
11. Noury N et al (2007) Fall detection-principles and methods. In: 2007 29th Annual international conference of the IEEE engineering in medicine and biology society. IEEE

12. Bourke AK, O'brien JV, Lyons GM (2007) Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. Gait Posture 26(2):194–199
13. Chen T et al (2015) Xgboost: extreme gradient boosting. R package version 0.4–2, pp 1–4
14. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM