Received 9 February 2023, accepted 21 February 2023, date of publication 27 February 2023, date of current version 1 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3249191

APPLIED RESEARCH

Fall Detection With Wrist-Worn Watch by Observations in Statistics of Acceleration

SONGSHENG LI
School of Computer Engineering, Guangdong Business and Technology University, Zhaoqing 526020, China
e-mail: songsheng.li@hotmail.com

This work was supported by Guangdong Business and Technology University.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Academic Committee of School of Computer Engineering, Guangdong Business and Technology University.

ABSTRACT It is common for older people to live alone, which can have tragic consequences if they have an accident and can't call for help in time. This is particularly acute in an aging society where falling is one of the most common accidents. According to the CDC, 1/4 of people over the age of 65 in the United States fall each year. The development of IoT and MEMS has made it possible to detect falls in time and automatically call for help. The presented fall detection system focuses on the walk-fall-still pattern, collects accelerations through the wrist-worn M5StickC-Plus watch, analyses the data locally in the watch, detects falls using an algorithm based on observations in the statistics of acceleration in one second, and then transmits the alarm signal to a remote healthcare system in real-time via WIFI. The lightweight algorithm has been proven to be 90% accurate in detecting falls, and the system can notify service staff of accidents within 1 second. The features of comfort, lightness, and timeliness make the device more practical than similar products. The low-cost, non-intrusive device can be used in care homes and is also suitable for elderly people living alone.

INDEX TERMS Fall detection, acceleration, wrist-worn, statistics.

I. INTRODUCTION

According to the report [1] from CDC (Centers for Disease Control and Prevention), more than 1 in 4 people over the age of 65 falls each year, and 1 in 5 falls result in serious injuries, such as broken bones and head injuries. In 2019, 3 million seniors across the United States went to the emergency room for a fall, and 34,000 seniors died as a result of a fall. Therefore, falls are a serious threat to the health of older adults. If an elderly person falls unexpectedly but is unable to get help in time, this can be a tragedy because untreated falls can aggravate the condition or even cause death. This problem is particularly acute for seniors who live alone or are alone for most of the day. A fall is a loss of balance. For the elderly, most of the external causes are related to the environment, such as lights, carpets, and floor, while the internal causes may be the effects of diseases and medications. Whatever the cause, if the elderly cannot regain their balance on their own to seek proper help, this could be the beginning of a tragedy. With today's highly advanced technology, is there a way to accurately and reliably identify accidental falls? Many researchers are using new devices and algorithms to identify falls and send out timely warning signals so that elderly people who have fallen can receive timely care and avoid tragedies.

Wang et al. [2] presented a literature survey of elderly fall detection systems, in the paper, they concentrated on sensors and investigated this topic comprehensively from multiple perspectives such as data collection, data transmission, data fusion, data analysis, security, privacy, and applications, etc. Data collection is mainly through 3 types of sensors, wearable sensors based on Accelerometer [3], [4], [5] and Gyroscope, vision sensors based on camera [6] and Kinect [7], and ambient sensors based on RF [8], Wi-Fi [9], Radar or Cellular, etc. Accordingly, the wearable is usually fixed around the waist, and the dominant algorithms are threshold and Machine Learning (ML), and for the latter two sensors, ML is the dominant algorithm. The summary is listed in Table 1. The advantages and disadvantages of both Threshold and ML are prominent, the former being simple and fast but relatively

The associate editor coordinating the review of this manuscript and approving it for publication was Chong Leong Gan.

TABLE 1. Various sensors applied in fall detection.

Type	Location	Sensors	Algorithm
Wearable	waist	Accelerometer and Gyroscope	Threshold/ML
Ambient	ambient	RF, Wi-Fi, Radar or Cellular	ML
Vision	indoor	Camera and Kinect	ML

low accuracy, and the latter being highly accurate but also computationally intensive, so sensor data fusion serves as a compromise and strikes a balance between accuracy and computational effort. Data fusion often uses information from two or more sensors to complement each other and increase the accuracy of results or the efficiency of calculations.

Some conclusions they came up with attracted attention, firstly, sensor data fusion [10] is mainstream, and its results are relatively stable but require high-intensity computation, so reducing the computational intensity of data fusion is one of the challenges of the topic. Secondly, many systems transfer data to the system server for analysis, not in the device, which affects response time and causes data security problems, so how to process data efficiently in real time becomes another challenge [11]. They also mentioned that there is no suitable standard dataset for evaluating and comparing the effectiveness of algorithms because the existing data sets [12], [13], [14] are simulated and not actual data on the behavior of elders.

Visual sensors will have the problem of privacy, and the corresponding algorithm is usually based on ML [15] whose high computational volume is not suitable for light-weight low-power processors. In contrast, for wearable sensors, mainly Accelerometer and Gyroscope, many previous studies have fixed the sensor in the waist [16] because the waist has a limited range of activities so the data changes significantly when falling. For person the most reasonable wearable device is watch alike, so the wrist is the most feasible location. The hands involve or with a variety of human actions, so the hand movements are the most frequent and diverse, from which distinguishing the fall is relatively more complicated. Fortunately, some researchers work on the area consistently with solid output.

A wrist-worn fall detection system using Accelerometers and Gyroscopes is proposed by Hsieh et al. [17]. They employ 2 devices on 2 hands, both with a tri-axis accelerometer and gyroscope, all the data collected from sensors will be transmitted to the computer by Zigbee. They set up a flow for detection, first, they check if the value from Gyroscopes is greater than 3500/s, then if at least one acceleration is higher than 6g, and a standard deviation of 0.4 seconds before or after the highest point must be less than 1.5, finally, they calculate the sum of acceleration of 2 seconds, if the sum less than 200g, the fall is true. They claim the average sensitivity and specificity of the system reach 95% and 96.7%. The way they treated the data in different sections is informative, but they do not explain how to combine data from the two devices because hands could be asynchronous while falling, and all the thresholds are empirical and need more clarification.

Interestingly, they gave up the wrist-worn way in the other paper [18], in which only one device is fixed around the waist. They propose a novel hierarchical fall detection algorithm including threshold-based and knowledge-based approaches. The absolute fall and activities of daily living (ADLs) are identified by the threshold-based classification, then the unidentified data frames flow to a knowledge-based fall detection algorithm, which first uses a multiphase fall model to segment the data, then features cover mean, standard deviation and skewness, etc. are extracted from signal to be fed to the SVM classifier, which is trained to classify the fall event into three distinct phases, freefall, impact and rest. They announce that the overall performances of sensitivity, specificity, precision, and accuracy were 99.79%, 98.74%, 99.05%, and 99.33%, respectively. It is close to perfect, but the sensor is moved from wrist to waist, it is a downgrade for a product in convenience, and the data is analyzed offline, not in the device. The algorithm waits 2.5 seconds for the rest data after impact, not counting the time for data transmission and classification, extra 2.5 seconds are delayed for the possible alarm.

Warunsin and Phairoh use a wristband with an ESP32 microcontroller to detect falls by Deep Learning [19]. TensorFlow [20] lite library is used to develop models to identify falls, the training dataset is MobiAct [12], and training and testing were performed on a laptop. They claim that the proposed model has an accuracy of 96.55%. Some issues are raised from the paper, first, they employ 128 sampling data points or 6.4 secs to predict activity, if half of them are after the fall, at least 3.2s delays of alarm are added. Secondly, the application of TensorFlow requires intensive computation and massive memory, this is the weakness of the microcontroller, although it does work, the battery will die quickly. The third is about the training dataset, from the description of MobiAct, data are measured by mobile phone in a pocket, not the wrist, the real data, and the training data from different sources, so the reliability of the result is not solid.

Khojasteh et al. [21] try to improve fall detection using an on-wrist wearable accelerometer, they reduce the computational constraints to embed the solution in smartphones or smart wristbands, and find that the rule-based systems with a reduced computational cost represent a promising research line as they perform similarly to neural networks, especially support vector machines performed with high specificity. de Quadros et al. [22] present a fall detection system for wrist wearable devices by movement decomposition and Machine Learning. Different sensors, signals, and direction components were combined and a comprehensive set of thresholds based and machine learning methods were applied to define the best approach for fall detection in the research. They get 91.1% accuracy for the threshold method and 99% for the machine learning method. Both methods have advantages and disadvantages, the choice depends on the application scenarios.

A ground-level fall is a fall in which the feet are on the ground, such as when standing, walking, or during the process of changing from seat to stand. The problem with the watch on the wrist is that the hands are flexible and not even synchronous with the trunk, when we reference fall, we mean the fall of the body, mainly the trunk, not the hands, but the hands fall with the body in the moment of fall.

The paper proposes a ground-level fall detection algorithm with a wrist-worn watch by observations in statistics of acceleration. The research concentrates on falls with the pattern of walk-fall-still, from walk to fall, then being still. Only two seconds of acceleration data are analyzed, one before the fall and the other after the fall, so it will be a one-second delay to alarm if a fall happens. A threshold of acceleration is applied as the trigger, for a typical fall, high acceleration implies touching down, then the context of the fall will be analyzed, the 1-second data before the threshold should be walk and fall, the 1-second data from the threshold moment is a touchdown and being still. If the 2-second data matches the pattern, it is a walk-fall-still fall. The decision of fall is based on the combination of a series of thresholds and ranges.

The research follows two main principles in the design of the fall detection algorithm. First, more conditions and loose threshold. The watch on the wrist is flexible and many unintentional actions could produce false falls, so more conditions are set to eliminate some ADLs, but less restriction on thresholds, as the elderly are vulnerable even if the fall is light but the damage could be serious. The second principle is lightness in the program, which means less computation and fewer resources used to conserve the battery, which in turn keeps the device working as long as possible.

In summary, the proposed algorithms, which are based on two basic principles, are characterized by lightweight procedures, fast 1-second responses, and data analysis within a wrist-worn device.

II. MATERIALS AND METHODS

The algorithm of fall detection will be elaborated in this part. The wrist-worn device, M5StickC Plus will be introduced first, followed by related concepts, such as quartiles and Standard Deviation (SD), then the workflow of the watch and core algorithms including freefall check and fall detection are explained.

A. AXIS DIRECTIONS

The device used in the project is M5StickC Plus [23], it is compact and portable with a size of 48 × 24 × 14mm but integrated with rich and powerful built-in hardware resources:

1) ESP32-PICO-D4 WITH WI-FI;
2) 6-AXIS IMU;
3) 120MAH LITHIUM BATTERY;
4) PASSIVE BUZZER;
5) 1.14-INCH LCD.

Just the parts related to the project are listed, our lightweight algorithm is run on an ESP32 processor; an alarm message will be sent out by Wi-Fi; the 3-Axis MEMS accelerometer is the data source of the project; Lithium battery and LCD make the device a real and practical watch, not just for experiments. The main concern is the data source, the 6-Axis MEMS IMU, which is a 3 mm x 3 mm x 0.75 mm 24-pin LGA package with a 3-Axis gyroscope and 3-Axis accelerometer, fixed inside the device, so the values of all 6 axis change with the movement of the hand. Only acceleration is employed in the project. The unit of acceleration values provided by the device is g, aka gravitational acceleration, which is 9.8ms^{-2}. 3-Axis exists in 3 dimensions, their actual directions relative to the surface of the device are shown in Fig. 1.

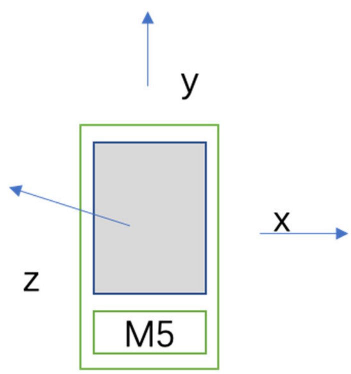

FIGURE 1. 3-Axis directions relative to the surface of the device.

From Fig. 1 we can see that the z Axis is vertical to the screen, the direction from the bottom to the screen is the positive z Axis, the text on the surface of the device from "M" to "5" direction is the positive x Axis, the direction from text "M5" to the screen is the positive y Axis.

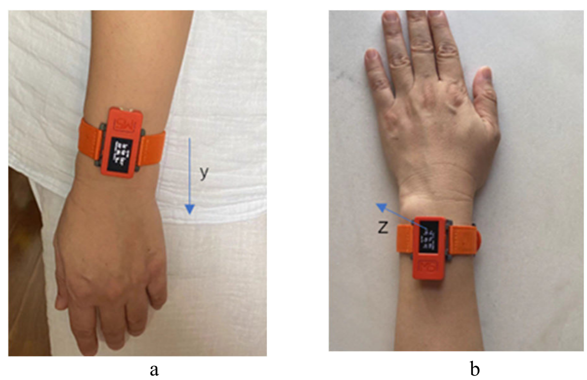

FIGURE 2. Two common states of rest with the watch, Stand (a) vs OnPlate (b).

Two common states of rest with the watch, Stand and OnPlate are compared in 3-Axis accelerations. The states are shown in Fig. 2 and the comparisons of 3-axis accelerations are shown in Fig. 3.

From Fig. 1 we know that the positive y-Axis is from text "M5" to the screen, so in Fig. 2 (a), the y-Axis is vertically downward in the state of Stand, accordingly, on the left side of Fig. 3, acceleration of y-Axis is about -1g, as we known acceleration of freefall is g, in this state of Stand, the watch is held still against the gravitational acceleration, so the acceleration is -1g.

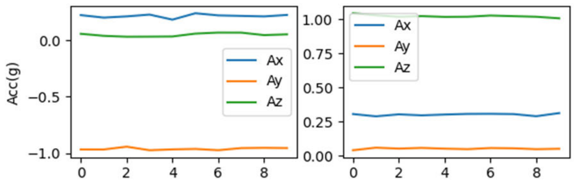

FIGURE 3. Accelerations of 3-Axis in Stand vs. OnPlate.

The positive Z-Axis acceleration in Fig. 1 is vertical to the screen and upward, so in Fig. 2 (b), in the state of OnPlate, the Z Axis is under the same condition as y Axis in the state of Stand except for the direction. On the right side of Fig. 3, the acceleration of z Axis is about 1g, all the support is concentrated on Z Axis and against the gravitational acceleration, but it is upward so the z Axis acceleration is +1g. On both sides of Fig. 3, Ax, acceleration in x Axis is about 0.25g because the watch is not exactly horizontal in Fig. 2.

B. RELATED STATISTICAL CONCEPTS

In the process of detecting falls, quartiles and Standard Deviation (SD) are applied to measure the dispersion of data. Quartiles are 3 values that divide the dataset into 4 groups evenly by quantity, they can be described in Box Plot intuitively. Specifically, a dataset with n items, marked as D[n], in ascending order, is separated into 4 identical parts, Q (1) is the value of the first quartile, or the 25th percentile, for $i = 1\sim3$, the position of Q(i) is:

$$P(i) = (n+1) * i/4 \tag{1}$$

And if P(i) is Integer,

$$Q(i) = D[P(i)] \tag{2}$$

or if P(i) is Float,

$$Q(i) = (D[floor(P(i))] + D[cell(P(i))])/2 \tag{3}$$

The implication of their names, floor (2.5) = 2 and cell (2.5) = 3, Interquartile Range (IQR) is the difference of Q (3) and Q (1), IQR = Q (3)-Q (1), and Q (2) is Median, upper limit equals Q (3) +1.5*IQR, and the lower limit is Q (1)-1.5*IQR, then all the other data in the dataset beyond Upper and Lower limit are called Outliers. Outliers play significant roles in the detection of falls as extreme values lead suspects of falls. Those definitions in Box are shown in Fig. 4.

IQR is the length of the Box, it represents the spread of the middle 50% of values in the dataset, called midspread as well.

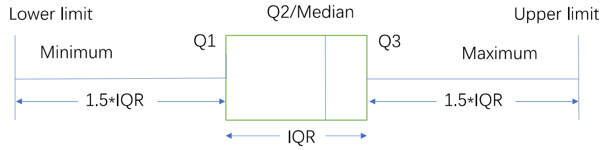

FIGURE 4. Definitions in box plot.

TABLE 2. Comparison of stand and walk in statistics.

Status	Stand	Walk
Mean	0.99	1.04
Median	0.99	1.05
IQR	0.01	0.18
SD	0.01	0.16

SD is the other typical measure of variation which can be calculated by:

$$\sigma = \sqrt{\sum_{i=1}^{n} (x_i - \mu)^2 / n} \tag{4}$$

μ is the mean of the dataset and σ is the average distance from the mean. Smaller SD means items in the dataset closer to the mean. From the formulas, we can see that SD is calculated by all the items in the dataset, and includes outliers, but IQR is decided by only the middle 50% of the dataset. In the proposed algorithm, SD or quartiles are used to assist judgment in some key points, if the dataset is not skewed, SD is employed, otherwise, quartiles are preferred.

SD is employed to analyze the states of the user, for instance, Fig. 5 is the comparison of the compositive acceleration (Ac) of Walk and Stand, Ac is calculated by:

$$\text{Ac} = \sqrt{Ax^2 + Ay^2 + Az^2} \tag{5}$$

Ax, Ay, and Az are accelerations in x-Axis, y-Axis, and z-Axis respectively. Fig. 5 shows that Ac for Stand is about 1g, g is the acceleration of gravity. Details of all 3 Axis are presented on the left of Fig. 3, but Ac for Walk swings from 0.7g to 1.3g.

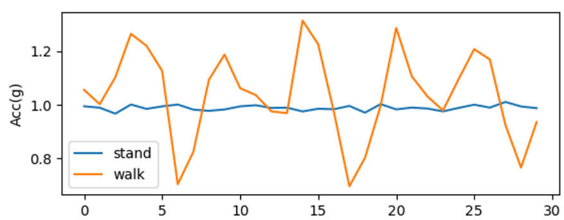

FIGURE 5. The comparison of the compositive acceleration (A_c) of Walk and Stand.

The differences between them are reflected in key statistical indexes, the values are shown in Table 2.

For both Stand and Walk, the Mean and Median are close, this proves that no outliers in both states, but IQR and SD of Walk is greater than Stand, which proves the swing of Walk and stability of Stand in acceleration. Besides Standing and walking, the feature of freefall is significant although the fall studied in this paper is not exactly the same as free-fall but the process is similar. The variety of accelerations of freefall in the device from a height of 1 meter is shown in Fig. 6.

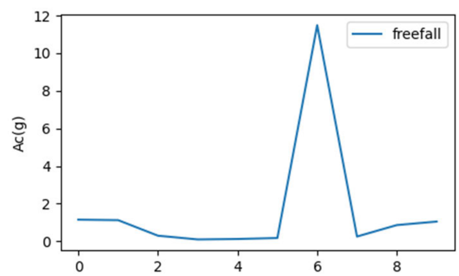

FIGURE 6. The wave of accelerations in freefall.

The data in Fig. 6 is collected in a second with a frequency of 10 Hz, so the unit of the x-axis is 1/10 second. At 0.1 seconds, the device is released from a height of 1 meter, it falls freely with no support so the acceleration from the device is 0g, then touches the ground at 0.6 seconds, with an acceleration of 12g. According to the equation of distance $d = 1/2gt^2$, as time is 0.5s, then $d = 0.5*9.8*0.5*0.5=1.225$ meter, not exactly 1 meter, the error is related to the frequency of data access. The pattern of acceleration from 1g to 0g and then 12g in freefall will help in the following fall detection algorithm.

The combination of BoxPlot and SD can reveal statistical details of the dataset while it is lightweight in computation and perfectly suited to be performed on a microcontroller like ESP32. A typical process of fall with pattern walk-fall-still, specifically from walk to fall, then being still, whose change of acceleration is shown in Fig. 7.

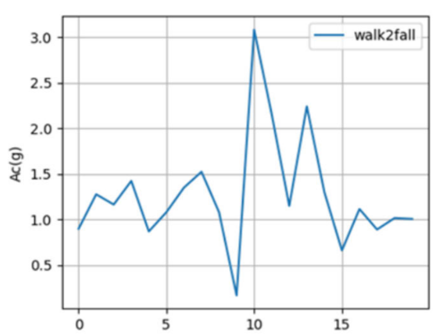

FIGURE 7. Acceleration of the process from walk to fall to still.

The figure includes the change of acceleration in all 3 phases, walk, fall, and still. The sampling frequency of acceleration is 10 Hz, 10 times per second, the x-axis is times, 20 times means 2 seconds, the data are separated into 2 parts, time 0 to time 9 is the left part, and time 10 to time 19 is the right part. In the left part, time 0 to time 8 is the state of a walk, at time 9 the Ac drops dramatically, it is the process of falling, similar to Freefall, then the right part, time 10 is the state of touching the ground, the Ac experience a change from minimum (Min) to maximum (Max), after that Ac keep vibrating but between the Min and Max, and becomes still gradually. The vibration is because the sponge mat is used as protection when experimenting. The first 8 times match the feature of Walk and the last 5 times match Still in Fig. 5, from time 8 to time 10 is the process of fall. The peak acceleration is just 3g, which is far from 12g of the real free fall, but the process of this typical fall is similar to freefall. If the feature of the right part is not Still, but Walk or similar to fall, the user is probably working out or falling then standing up quickly.

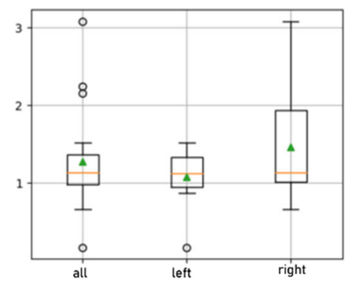

FIGURE 8. Acceleration of the process of walk-fall-still in BoxPlot.

Fig. 8 is the boxplot of 3 group data, all, left part and right part. When the data is treated as a whole, the Max and Min are outliers, the Min in the left part is still an outlier but Max is not an outlier in the right part. Outliers could be used to judge for fall. The key statistical indexes of fall are listed in Table 3. The values will be employed as a reference for the following algorithms.

TABLE 3. Key statistical indexes of fall.

	All	Left	Right
Mean	1.27	1.08	1.46
Median	1.13	1.12	1.13
IQR	0.43	0.32	0.72
SD	0.61	0.37	0.73

The main feature of fall is a Min Ac followed by a Max Ac. This feature trigs a possible fall judgment, the final decision depends on the following status of the user, for being still, it is probably a fall, otherwise, not. And the feature of Still, Walk, or Workout is judged by statistical indexes. These will be explained in the following algorithms.

C. WORKFLOW OF THE DEVICE

As mentioned before the device employed in the system is M5StickC PLUS, and the integrated development environment used in the project is Arduino, all programs of Arduino are running on two functions, one is Setup(), which will be executed once if the device is restarted or reset so it is suitable for initialization; the other function is Loop(), which runs repeatedly like the implication of its name so all the routine codes are located inside. The workflow of fall detection in the device is shown in Fig. 8.

Based on the second principle of design, lightweight, the program just applies two integer variables and a float array of 10 elements from the beginning, one integer variable is index, which is the current writing position for the circular list made by the float array. The circular list is designed to promise that the last ten values of the Ac are kept. This is described on the left side of Fig. 8, the index is initialized as 0, the first Ac will be written to Ac[0], then index plus 1, so the second Ac will be written to Ac [1], the circular list keep going until index reaches 10, it means the array is full, it keeps the last 10 Ac-s, the oldest is Ac[0], the newest is Ac [9], so when the next one comes, Ac[0] should be replaced, this is why on the left side of Fig. 8 when the index reaches 10, it returns to 0, then the circle starts over. In this way, when the index is 5, the oldest Ac is Ac [5] and the newest is Ac [4], the sequence from oldest to newest is Ac [5] to Ac [9], then Ac[0] to Ac [4], always 10 last Ac values. The process is shown on the left side of Fig. 9.

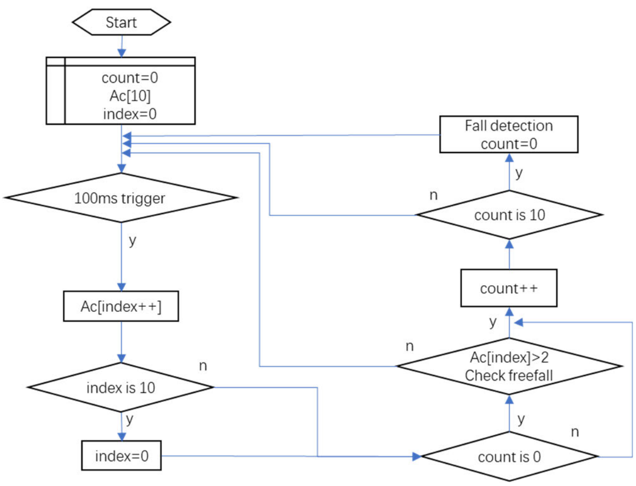

FIGURE 9. **Workflow of the device.**

The other integer variable is count, which is set to 0 at the beginning too, it is the counter for incoming Ac-s if the following 2 conditions are met, one is the threshold of Ac, which is set to 2g, it is smaller compared to the other algorithm [17], they use 3g or 4g as the threshold. The setting meets the first design principle. The other is that freefall is found in the last 5 Ac-s. If the conditions are satisfied, the counter starts, then counter pluses one on every 100ms. Totally 10 Ac-s are collected, and those data will be applied to the fall detection algorithm. No matter if it is a fall or not, after 1 second the counter is reset to 0, which means it is ready for the next round of detection. The procedure of conditions checks and counter is shown on the right side of Fig. 9.

Index and counter work separately, without interference. They both are from 0 to 10, but index is continuous and counter has to wait for the satisfaction of the conditions. The two key algorithms, which are check of freefall and fall detection, referring to Fig. 7, check of freefall works on data of the left part, from time 0 to 9; fall detection employs data of the right part, from time 10 to 19, they will be explained in more detail in the following. Their workflows are shown in Fig. 10 and Fig. 11.

D. CHECK OF FREEFALL

The step before checking for freefall is preparing data. The last 10 Ac-s are retrieved from the circular list in time order, from the oldest to the newest. The check includes two steps, the first is calculating the SD of the last 10 Ac-s to make sure that the user is in the state of Walk as the detection pattern is walk-fall-still. Technically, taking results from Table 2 for consideration, SD should be more than 0.1, and from Table 3, the left side data from 0 to 9, its SD is 0.37, so the range for SD is set to [5, 0.1], to avoid being still and workout.

The second step is finding a freefall sign in the last 5 Ac-s. For freefall, the Ac should be zero from experience in Fig. 6, but practically the device is attached to the wrist, it is droved and supported by hand, so Ac is only close to zero given the short time and short distance. Reference from Fig. 5, the Ac of a normal walk is more than 0.6g.

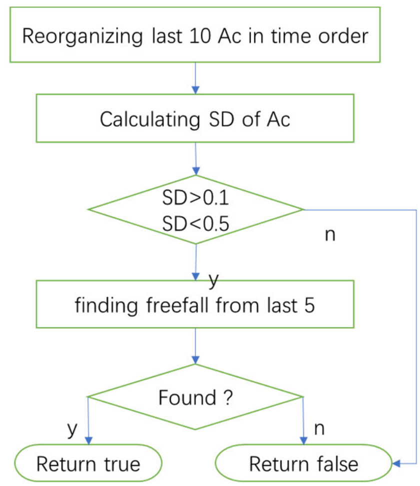

FIGURE 10. **Workflow of check freefall.**

The threshold of Ac for fall is set to 0.5g. The freefall should be found in the last 5 Ac-s, as the interval of the trigger is 100ms, 5 Ac-s is half a second, it is the empirical value from freefall of 1 meter which is more than half the height of a normal person, so is reasonable for the condition. The flow of this freefall check is shown in Fig. 10.

E. FALL DETECTION

According to the focused pattern of fall, walk-fall-still, walk, and fall are confirmed in check of freefall. The last task is confirmation of being still. First of all, the 10 last Ac-s are reorganized from the circular list in time order, those data are the right part in Fig. 7. Ac[0] is the oldest and Ac [9] is the newest, then Ac [9] is checked to decide if the state of being still can be denied, the range is [0.8g,1.2g]. If Ac is far from 1g, it means the user is acting, so it is not a fall needs alarm. More evidence needs to prove that the user is in the state of still, so the Ac-s are iterated from Ac [9] to Ac[0] to find how many continuous Ac-s less than 1.5g, marked as p. If p<5, this cannot be detected as fall for two reasons, one is no enough data support that the user is still, the other is the user probably move in the first part of this second as Ac>1.5g. If p>5, the SD of those p Ac-s is calculated, it must be small to prove the state of stillness. Being still, the SD should be close to 0, but in experiments using a sponge mat, the rebounds exist, and the empirical value is set to 0.2. If all the conditions are met, the decision of fall is true, an alarm will be sent out. There are 3 thresholds in the block, Ac, time (p), and SD for being still. The workflow of fall detection is shown in Fig. 11.

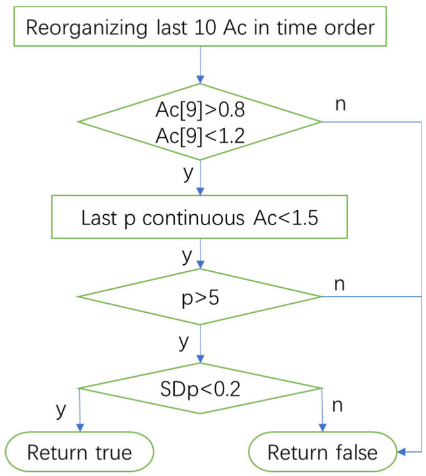

FIGURE 11. Workflow of fall detection.

In conclusion, the simple version of the whole fall detection process is shown in Fig. 12.

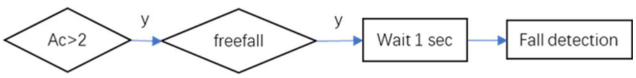

FIGURE 12. Workflow of the whole process.

The pattern of fall in the research is walk-fall-still. If the threshold of Ac is met, fall detection is trigged, the last 10 Ac-s are checked for a walk, and the last 5 Ac-s are checked for freefall, if they are satisfied, the program waits for a second for 10 new Ac-s to decide fall or not. Totally 20 Ac-s need, 10 before the possible fall and 10 after the possible fall, the same as the descriptions in Fig. 7.

III. RESULTS

This section begins with a description of the experiment setup and experimentation process, followed by an explanation of the evaluation criteria and the impact of daily activities (ADLs) on fall detection. The algorithmic accuracy is then discussed.

A. EXPERIMENTS SETUP

The device, M5StickC Plus is worn on the wrist of the left hand, the screen is on the back of the hand and the positive y-axis is in the same direction with the fingers, which is shown in Fig. 2a. In the experiment, there is a 2-meter type-c data cable connecting the device and computer to provide real-time measurement results. The sample rate of acceleration is 10 Hz, 10 times per second, and the interval for a sample is 100ms.

Other tools include a sponge training mat and a chair with arms. The tools are shown in Fig. 13.

FIGURE 13. Auxiliary experimental tools.

B. THE PROCEDURE OF THE EXPERIMENT

One of the distinctive features of our study is the device, the watch on the wrist, its flexibility and mobility could confuse detection. When designing of experiment procedure for fall detection, it is important to try actions that could create a similar data sequence to the real fall. Based on different tools, the procedure is divided into 2 parts, one part is on a chair, simulating some actions that may cause misdetection, and the other is on a sponge mat, performing real falls forward and backward. The procedure is shown in Fig. 14.

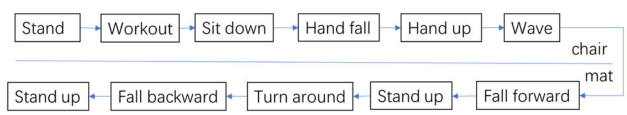

FIGURE 14. The procedure of the experiments.

To explain why some actions like sit-down, hand-fall, and hand-wave are selected, a typical Ac-s for the whole experiment procedure is shown in Fig. 15. The wave is divided into sections A to G in the order of the procedure. Section A is a workout, like doing exercises, hands up and down, left and right, for participants, this is the freestyle stage. From the Fig. 15 we can see some parties look like a feature of fall, local minimum then local maximum, similar to freefall then a touchdown, it can trig the detection, but the concentration of the research is walk-fall-still, the algorithms check all 3 parts, so workout does not make much trouble for the algorithms in general. Section B is the action of sit-down, the wave looks like a fall as well, but the trigger is Ac>2g, and the walking part is not qualified. The action of section C is that the left hand drops from the handle of the chair naturally, the wave is very close to a typical fall except for the walking part, which is the most possible part to be mis-detected. Hanging is not comfortable, soon left-hand lifts up and back to the arm of the chair, this is Section D. It does not meet the fall part of the algorithm. Section E is the hand wave then back to the arm of the chair, this could happen when sitting in a chair, but for detection, there is no freefall part. Section F and G are typical walks to fall forward and backward, then being still, the difference is when falling backward hand could act more than forward as users try to get some support.

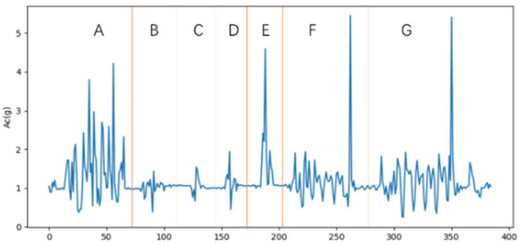

FIGURE 15. The typical Ac for the whole procedure of the experiments.

Ten young participants are invited to join the experiment, 7 males and 3 females, they are all students, aged from 19 to 22. They wear the device on the left wrist, the device connecting to the computer by Type-c cable, so the results of experiments are shown on the computer in real-time by the serials monitoring provided by Arduino, mainly fall alarms from the watch, and all the data are saved in the computer for further analysis. Participants are suggested to keep 2 or 3 seconds still between actions to make the boundary clear. All the participants follow the procedure and experiment 5 times, so 50 group data like Fig. 15 are used to evaluate the performance of the fall detection algorithm.

C. EVALUATION CRITERIA

The result of the core algorithms is always a true or false fall, it belongs to the area of binary classification, the common evaluation criteria of which is the confusion matrix [25]. The structure of the confusion matrix is shown in Table 4.

TABLE 4. Structure of confusion matrix.

		Prediction	
		Positive	Negative
Truth	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

The meaning of the four elements TP, FP, FN, and TN can be explained by the experiments data, for instance, taking 10 groups data for consideration, there are 10 tries of fall forward, if the device reports 8 falls, then TP=8, FN=2, in the meantime, 10 tries of hand-wave report 1 fall, then FP=1, TN=9. So FP and FN are the numbers that the algorithms try to push to zero. Multiple accuracy metrics are defined based on the confusion matrix. Calculation and value from the example of metrics are listed in Table 5.

TABLE 5. Calculation and value of metrics.

Metric	Definition	Value
Sensitivity	TP/(TP+FN)	0.8
Specificity	TN/(FP+TN)	0.9
Precision	TP/(TP+FP)	0.89
Accuracy	(TP+TN)/(TP+FP+TN+FN)	0.85
F1 score	2*Precision*Sensitivity/ (Precision+Sensitivity)	0.84

From the definitions, we can see that the calculation of metrics needs only the sample of two actions in the same numbers. Accuracy is useful for reference, so it is suitable for our experiments as fall and hand-wave are in every trial. The F1 score is a combination of Precision and Sensitivity, so it is more informative. Overall, a higher value of those metrics means better performance of the algorithms. The metrics will be employed to evaluate the performance of the algorithm.

D. RESULTS OF THE EXPERIMENTS

The collection of results in the experiments is simple, as mentioned before, the device connects to the computer by a Type-c cable to send messages to serial monitoring of Arduino. It is called a log. A section of data with fall detection in the log is shown in Fig. 16.

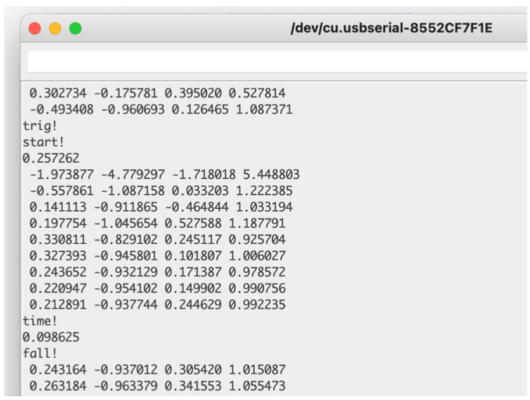

FIGURE 16. A section of data with fall detection in the log.

Every row of the normal data in the log has 4 columns, they are Ax, Ay, Az, and Ac respectively. And the promotion of key events is added to the messages, such as the Ac threshold is trigged, fall detected and the related SD value, they are beneficial to debug.

For result collection, every time participants finish a round of procedure, the messages from the serial log are saved and drawn similarly to Fig. 15, if any fall messages are presented in the log, they are marked manually with the corresponding actions.

As mentioned before, 50 groups of data on the procedure are collected. The total recorded falls from different actions are listed in Table 6.

TABLE 6. Distribution of falls with actions.

Action	Fall	Total
Workout	1	12
Hand fall	6	
Hand up	2	
Wave	3	
Fall forward	47	92
Fall backward	45	

If all the results are correct, there should be 100 falls, 50 falls forward and the same backward. From the results, a total 104 times of falls are reported, as analyzed before, the most possibly led to mis-detected actions are hand-fall and hand-wave, the result reflected the theory basically, but Workout making false fall is unexpected. When checking the data, it seems the Workout in the experiment is a little tender and slow. We separate the truth as fall and not fall, and get the confusion matrix in Table 7.

TABLE 7. Confusion matrix of results.

		Prediction	
		Positive	Negative
Truth	100	TP=92	FN=8
	100	FP=12	TN=88

As a result, the metrics are shown in Table 8.

TABLE 8. Metrics of the experiments.

Action	Fall
Sensitivity	0.92
Specificity	0.88
Precision	0.88
Accuracy	0.9
F1 score	0.9

Accuracy and F1 score are the same, 90%, and Sensitivity is slightly higher than Specificity as TP is higher than TN. In the experiments, M5StickC Plus is employed as a watch on the left wrist of participants who do experiments following a designed procedure of some ADLs and two common ways of fall, the results are evaluated by a confusion matrix, and the accuracy of fall detection is about 90%. The result will be discussed in the next section.

IV. DISCUSSION

In this section, the comparison with other algorithms based on wrist-worn devices is presented first, followed by the consideration of thresholds and ranges, the issue of experiments, and finally the future direction of research.

A. COMPARISON WITH OTHER ALGORITHMS OF WRIST-WORN DEVICES

The main concern about the system is its accuracy. From the result of the last section, the experiments based on 50 round procedures reach an accuracy of 90%. The comparison with other algorithms based on wrist-worn devices is listed in Table 9.

TABLE 9. Comparison with other algorithms based on wrist-worn devices.

Authors	Hsieh et al. [17]	Warunsin et al. [19]	proposed
Year	2014	2022	2022
Sensors	Accelerometer& Gyroscope	Accelerometer	Accelerometer
Methodology	Threshold	Deep Learning	Threshold& range
Accuracy	95%	96.55%	90%
On-site result	No	Yes	Yes
Length of data	2.8s	6.4s	2s
Alarm delay	>2.4s	>3.2s	>1s

To the best of my knowledge, wrist-worn devices are not the first choice for fall detection schemes because of the flexibility and freedom of hands, but as mentioned before, the watch is the device that the user is used to and does not feel redundant. The flow of Hsieh et al. [17] started with Gyroscope, they defined a threshold for Gyroscope in y-axis and z-axis. We find that the values of Gyroscope are too volatile to be applied as a threshold. Both Hsieh et al. [17] and the proposed algorithm use thresholds to detect falls, the difference is that Hsieh et al. [17] transmit collected data to the computer through Zigbee, and the analysis of data is executed in an extra computer, but the proposed algorithm is carried out in the device. The decision of Warunsin et al. [19] is made in the device as well, but the training of the model is run on an extra computer. The advantage of calculation on-site is that the device does not transmit data to outside devices, the process of transmission could exhaust the device. Only the proposed algorithm does not depend on extra devices. The other advantage of the proposed algorithm is response time. The length of data in Table 9 is the length of time for the necessary data of one detection in seconds. Hsieh et al. [17] employed 0.4s data before possible fall and 0.4s after that, then 2s after, a total of 2.8s data. Warunsin et al. [19] demand a number of 128 sampling data points or 6.4 seconds period to predict activity, the proposed algorithm need only 2s data, 1s before fall and 1s after to detect fall. Alarm delay includes two parts, one part is from fall to detection of fall, Hsieh et al. [17]

need 0.4s+2s=2.4s, and Warunsin et al. [19] provide 6.4s data for deep learning but no details about the trigger for the data, so average value 6.4s/2 is used. The proposed algorithm needs 1s after the possible fall. The other part of alarm delay is from the detection of fall to the system sending out an alarm message, Hsieh et al. [17] do not provide related information, and Warunsin et al. [19] employ LINE, the proposed algorithm sends out an alarm message by a health sensing system [26], [27]. The delay of alarm must be more than 2.4s, 3.2s, and 1s respectively.

Compared to other published algorithms claimed more than 90%, this result is relatively lower. This is the weakness of the proposed algorithm, but the advantage is evident too. First of all, the judgment of fall is made in the device. Our target is to make the device a real practical product, it does not depend on the auxiliary device to make a decision. Secondly, the response time is the shortest. Time is life, we manage to send out an alarm message as soon as possible. Thirdly, energy efficiency is a priority. When designing the algorithm, the most complicated computation is SD. Transmission is exhausted, so only the necessary alarm message is sent out, not the original data. Finally, threshold and ranger are adjusted to meet the situation of the elderly although the young are invited to the experiments. We insist on using the watch because it is the most convenient and reasonable way, but we narrow down the fall detection to the pattern of walk-fall-still, three parts make the algorithm reliable.

B. THRESHOLDS AND RANGES

Back to the algorithm, thresholds and ranges are used to guide the workflow. The first threshold is Ac>2g in Fig. 9, this is the trigger for fall detection. It is relatively low compared to other algorithms [17], [18], [19]. The threshold was deliberately lowered in consideration of the fact that the ultimate service recipients are the elderly. Even light fall hurts. This setup causes some issues which can be seen in Fig. 15, section C, hand fall, and section D, hand up can meet the requirement easily, in turn, increase the FP value and decrease the accuracy. On the contrary, if the threshold is raised, FP will drop but TP could drop too as for some light falls the touchdown Ac does not meet the threshold. In the experiments, as all the participants are young students, raising the threshold should decrease FP and not affect TP, then increase accuracy. So, this dilemma is the art of balance.

The first range is the SD range of walk, which is defined as [0.1,0.5]. From Table 2, we can see that the typical SD for standing is about 0.01, and for walking is 0.16, so the lower limit is set to 0.1. From Fig. 7 and Table 3, the SD of the typical left side data is 0.37, so the upper limit of SD for the walk is set to 0.5, the other basis of this decision is the SD of Section A, workout in Fig. 15 is 0.73, so 0.5 can exclude the workout. In our experiments, there is 1 FP from the workout, it is related to the range.

The second threshold is the Ac for freefall. Theoretically, Ac for freefall is close to 0 from Fig. 6, based on observation from Fig. 5, Ac is no less than 0.6g for a normal walk, so 0.5g is chosen as the threshold for freefall. This threshold could cause FP as well, for example, the feature of the hand-wave (Section E in Fig. 15) is without fall before the trigger, and it does not match the walk-fall-still pattern, but in the experiments, there are 3 FP from the hand-wave because of this. But if the threshold is set lower, some true fall could be missed because Ac is not low enough to meet this threshold. The choice of thresholds 1 and 2 fully reflects the first principle, loose threshold, for threshold 1, the Ac should be very high, we lower the threshold; for threshold 2, the Ac should be low, so we raise the threshold.

The second range is to decide if the user is in the state of still, the Ac should close to 1g, so it is set as [0.8,1.2]. The third threshold is the value of p, it is the time for the user being still, 5 means half a second. From Fig. 7 we can see the 1-second data on the right side, half second change dramatically half second being still. This is why p is set to 5. Technically, Touchdown following Fall, the wave of Ac-s should become still quickly, actually, the half-second dramatic waves are the rebounds from the sponge mat which we used in the experiments. We could use knee and elbow pads, but no appropriate protection for the hip, so a sponge mat is a better choice. The mat brings out the other issue that for some true falls the time of being still is less than half a second, this is where FN from. This is the main reason that Sensitivity is only 92%. One way to improve the result is to wait one more second or at less half a second so the still will be confirmed, but the 1-second response time of the fall alarm has to be changed to 2 seconds or 1.5 seconds. The other way is to raise the sampling frequency, the current is 10 Hz, 10 times a second, for 20 Hz, the data will certainly have more details with 20 Ac-s in a second, but we can already see the full picture by 10 Hz, and higher sample rate will result to faster battery drain of the device.

The last threshold is SD for being still, it should be close to 0, but affected by the waves caused by the sponge mat, 0.2 is a value of compromise.

C. ISSUES OF THE EXPERIMENTS

As mentioned before 10 young students perform the series of actions from the designed procedure 5 times, and results are summed up manually based on the log from the serial of the device. Protection for the fall is a sponge mat for sport. The setup brings an obvious issue that the data is not really from the elderly, but this is the reality, even some open datasets of falls can provide none or a small group of data from the elderly. FAR-SEEING [12] (Fall Repository for the design of Smart and self-adaptive Environments prolonging independent living) is a dataset of falls in real life, but the falls of the elderly are from vision sensors. UMAFall [13] and DaLiac [24] include data from acceleration on the wrist but do not mention any from the elderly.

In our research, some thresholds and ranges are intentionally lowered to meet the requirement of the elderly. First of all, the first threshold of Ac as the trigger for fall detection is set to 2g. Only slight movement can reach 2g of Ac, this is the first principle of design, more conditions, loose thresholds. When filtering walks, the SD is lowered to 0.1 as seniors could walk tenderly and slowly. And the Ac for fall is set at a high value of 0.5g based on the same reason. In conclusion, we know the experiment is a simulation, and the algorithm has been designed to take into account the deviation of data as much as possible.

The other issue is caused by the sponge mat, exactly the rebounds from the sponge result in the dramatic change of Ac-s in the first half second after the fall, in turn, causes the increase of FN and decrease in Accuracy. Fig. 6 is the freefall, almost no rebounds appear after the fall because a book is put on the sponge mat for the device, fall on the normal ground should be similar.

We emphasize that the device is a real practical product, but in the experiment, we found that the fully charged device working on the algorithm can only last about 2 hours because the volume of the battery is only 120mAh. The full plan of the system is that when falls are detected the devices search and connect to a platform by Wi-Fi or Bluetooth to send an alarm message to the related responder. This part consumes the battery heavily too.

D. FUTURE RESEARCH DIRECTIONS

In response to the previously mentioned issues, some improved solutions could be feasible.

First of all, research on this topic will be consistent on the employment of the watch on the wrist because it is the natural and convenient way without intrusion. Currently, the device is a general development device, the physical interfaces and the trademark occupy 1/3 of the surface, for a real product this 1/3 will be taken and the battery of volume will be increased. The normal battery capacity of the Apple Watch is 250 mAh to 300 mAh, if the battery of the device can be doubled, not only the time of endurance we can also try some other algorithms.

TensorFlow Lite Micro is announced to support the ESP32 in 2020. It introduces Machine Learning and even Deep Learning algorithms into the device, but the requirement of heavy computation and massive memory for TensorFlow is still a huge challenge for a watch with a limited battery.

The research concentrates on the fall pattern of walk-fall-still, but the other pattern of fall is common for the elderly, for example, stand up but fall. The research makes every effort to eliminate this pattern as the walk is checked first in the algorithm, but as a general device, only detecting falls of a certain pattern is not good enough, making the device versatile in fall detection is another future research direction.

We develop a health sensing system [26], [27] for Smart pillow in other research and ESP32 works as an agent in the system between pillows and the platform. The watch can access the platform by the ESP32 agent using Wi-Fi or Bluetooth too, the watch will expand and enrich the application range of the system.

V. CONCLUSION

In this paper, a ground-level fall detection system is presented, which uses a wrist-worn watch as the hardware, and detects falls on observations in statistics of acceleration. The system concentrates on the pattern of walk-fall-still and has the following features.

First, the product is practical, it is a watch with a rechargeable battery and screen, compared with some big brands, the product has the advantages of low cost and a dedicated function for fall detection.

Secondly, the algorithm is lightweight. There is no heavy computation in the algorithm, no extra algorithm library is used, only the basic statistical concepts are employed, and the most complicated calculation is SD. This can keep the watch working as long as possible.

Thirdly the response time is 1 second. From the fall detection being trigged to the final judgment, there is only a 1-second delay, to the best of my knowledge, this is the fastest response in the fall detection research employed watch.

Finally, the algorithms are designed with the elderly in mind, with key thresholds and ranges optimized for the elderly, and the ultimate goal of this project is to integrate the device into the health sensing system we have previously designed for nursing homes.

The prototype of the watch is designed as a development kit, if it is reshaped to be square and its battery capacity increased, it will be a practical product.

REFERENCES

[1] *Older Adult Falls | Fall Prevention | Injury Center | CDC*. Accessed: Oct. 10, 2022. [Online]. Available: https://www.cdc.gov/falls/index.html

[2] X. Wang, J. Ellul, and G. Azzopardi, "Elderly fall detection systems: A literature survey," *Frontiers Robot. AI*, vol. 7, p. 71, Jun. 2020, doi: 10.3389/frobt.2020.00071.

[3] M. Saleh and R. L. B. Jeannès, "Elderly fall detection using wearable sensors: A low cost highly accurate algorithm," *IEEE Sensors J.*, vol. 19, no. 8, pp. 3156–3164, Apr. 15, 2019, doi: 10.1109/JSEN.2019.2891128.

[4] O. Kerdjidj, N. Ramzan, K. Ghanem, A. Amira, and F. Chouireb, "Fall detection and human activity classification using wearable sensors and compressed sensing," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 1, pp. 349–361, Jan. 2020, doi: 10.1007/s12652-019-01214-4.

[5] T. Wu, Y. Gu, Y. Chen, Y. Xiao, and J. Wang, "A mobile cloud collaboration fall detection system based on ensemble learning," 2019, *arXiv:1907.04788*.

[6] Q. Han, H. Zhao, W. Min, H. Cui, X. Zhou, K. Zuo, and R. Liu, "A two-stream approach to fall detection with MobileVGG," *IEEE Access*, vol. 8, pp. 17556–17566, 2020, doi: 10.1109/ACCESS.2019.2962778.

[7] A. Shojaei-Hashemi, P. Nasiopoulos, J. J. Little, and M. T. Pourazad, "Video-based human fall detection in smart Homes using deep learning," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5, doi: 10.1109/ISCAS.2018.8351648.

[8] Y. Tian, G. H. Lee, H. He, C. Y. Hsu, and D. Katabi, "RF-based fall monitoring using convolutional neural networks," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technologie*, vol. 2, no. 3, p. 137, Sep. 2018, doi: 10.1145/3264947.

[9] *KS-FALL: Indoor Human Fall Detection Method Under 5GHz Wireless Signals IOPscience*. Accessed: Dec. 14, 2022. [Online]. Available: https://iopscience.iop.org/article/10.1088/1757-899X/569/3/032068

[10] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, and C. Peñafort-Asturiano, "UP-fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, p. 1988, Apr. 2019, doi: 10.3390/s19091988.

[11] C. Medrano, R. Igual, I. Garcia-Magariño, I. Plaza, and G. Azuara, ''Combining novelty detectors to improve accelerometer-based fall detection,'' *Med. Biol. Eng. Comput.*, vol. 55, no. 10, pp. 1849–1858, Mar. 2017, doi: 10.1007/s11517-017-1632-z.
[12] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, ''The MobiAct dataset: Recognition of activities of daily living using smartphones,'' in *Proc. Int. Conf. Inf. Commun. Technol. Ageing Well e-Health*, 2016, pp. 143–151, doi: 10.5220/0005792401430151.
[13] J. Klenk, L. Schwickert, L. Palmerini, S. Mellone, A. Bourke, E. A. F. Ihlen, N. Kerse, K. Hauer, M. Pijnappels, M. Synofzik, K. Srulijes, W. Maetzler, J. L. Helbostad, W. Zijlstra, K. Aminian, C. Todd, L. Chiari, and C. Becker, ''The FARSEEING real-world fall repository: A large-scale collaborative database to collect and share sensor signals from real-world falls,'' *Eur. Rev. Aging Phys. Activity*, vol. 13, no. 1, p. 8, Dec. 2016, doi: 10.1186/s11556-016-0168-9.
[14] E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, ''UMAFall: A multisensor dataset for the research on automatic fall detection,'' *Proc. Comput. Sci.*, vol. 110, pp. 32–39, Jan. 2017, doi: 10.1016/j.procs.2017.06.110.
[15] Y. Kong, J. Huang, S. Huang, Z. Wei, and S. Wang, ''Learning spatiotemporal representations for human fall detection in surveillance video,'' *J. Vis. Commun. Image Represent.*, vol. 59, pp. 215–230, Feb. 2019, doi: 10.1016/j.jvcir.2019.01.024.
[16] A. Sucerquia, J. López, and J. Vargas-Bonilla, ''Real-life/real-time elderly fall detection with a triaxial accelerometer,'' *Sensors*, vol. 18, no. 4, p. 1101, Apr. 2018, doi: 10.3390/s18041101.
[17] S.-L. Hsieh, C.-C. Chen, S.-H. Wu, and T.-W. Yue, ''A wrist -worn fall detection system using accelerometers and gyroscopes,'' in *Proc. 11th IEEE Int. Conf. Netw., Sens. Control*, Apr. 2014, pp. 518–523, doi: 10.1109/ICNSC.2014.6819680.
[18] C.-Y. Hsieh, K.-C. Liu, C.-N. Huang, W.-C. Chu, and C.-T. Chan, ''Novel hierarchical fall detection algorithm using a multiphase fall model,'' *Sensors*, vol. 17, no. 2, p. 307, Feb. 2017, doi: 10.3390/s17020307.
[19] K. Warunsin and T. Phairoh, ''Wristband fall detection system using deep learning,'' in *Proc. 7th Int. Conf. Comput. Commun. Syst. (ICCCS)*, Apr. 2022, pp. 223–227, doi: 10.1109/ICCCS55155.2022.9846023.
[20] C. Chan, I. Verdiesen, J. Carvajal-Godinez, and P. S. Mani, ''TensorFlow TM open source library for machine learning applications,'' in *TU Delft Students on Software Architecture*. Lyon, France: GitBook, May 2016.
[21] S. Khojasteh, J. Villar, C. Chira, V. González, and E. de la Cal, ''Improving fall detection using an on-wrist wearable accelerometer,'' *Sensors*, vol. 18, no. 5, p. 1350, Apr. 2018, doi: 10.3390/s18051350.
[22] T. de Quadros, A. E. Lazzaretti, and F. K. Schneider, ''A movement decomposition and machine learning-based fall detection system using wrist wearable device,'' *IEEE Sensors. J.*, vol. 18, no. 12, pp. 5082–5089, Jun. 2018, doi: 10.1109/JSEN.2018.2829815.
[23] *M5StickC PLUS*. Accessed: Dec. 10, 2022. [Online]. Available: https://docs.m5stack.com/en/core/m5stickc_plus
[24] H. Leutheuser, D. Schuldhaus, and B. M. Eskofier, ''Hierarchical, multi-sensor based classification of daily life activities: Comparison with state-of-the-art algorithms using a benchmark dataset,'' *PLoS ONE*, vol. 8, no. 10, Oct. 2013, Art. no. e75196, doi: 10.1371/journal.pone.0075196.
[25] A. Kulkarni, D. Chong, and F. A. Batarseh, ''Foundations of data imbalance and solutions for a data democracy,'' in *Data Democracy*, F. A. Batarseh and R. Yang, Eds. New York, NY, USA: Academic, 2020, pp. 83–106, doi: 10.1016/B978-0-12-818366-3.00005-8.
[26] S. Li and C. Chiu, ''A smart pillow for health sensing system based on temperature and humidity sensors,'' *Sensors*, vol. 18, no. 11, p. 3664, Oct. 2018, doi: 10.3390/s18113664.
[27] S. Li and C. Chiu, ''Improved smart pillow for remote health care system,'' *J. Sensor Actuator Netw.*, vol. 10, no. 1, p. 9, Jan. 2021, doi: 10.3390/jsan10010009.

SONGSHENG LI received the B.S. degree from the Beijing University of Post and Communication, in 1995, the M.S. degree from Xi'an Jiaotong University, China, in 2002, and the Ph.D. degree from the University of Technology Sydney, Sydney, in 2013. He is currently an Associate Professor with the School of Computer Engineering, Guangdong Business and Technologies University. His research interests include wireless sensor networks, the IoT, UAVs, smart homes, and smart cities.