



# Real-time human posture recognition using an adaptive hybrid classifier

Shumei Zhang<sup>1</sup> · Victor Callaghan<sup>2</sup>

Received: 8 October 2019 / Accepted: 1 August 2020 / Published online: 12 August 2020  
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

## Abstract

A reliable adaptive hybrid classifier (*hAHC*), which combines a posture-based adaptive signal segmentation algorithm with a multi-layer perceptron (MLP) classifier, together with a plurality voting approach, was proposed and evaluated in this study. The *hAHC* model was evaluated using a real-time posture recognition framework that sought to identify five behaviours (sitting, walking, standing, running, and lying) based on simulated crowd security scenarios. It was compared to a single MLP classifier (*sMLP*) and a static hybrid classifier (*hSHC*) from three perspectives (classification precision, recall and F1-score) that used the real-time dataset collected from unfamiliar subjects. Experimental results showed that the *hAHC* model improved the classification accuracy and robustness slightly more than the *hSHC*, and significantly more compared to the *sMLP* (*hAHC* 82%; *hSHC* 79%; *sMLP* 71%). Additionally, the *hAHC* approach displayed the real-time results as animated figures in an adaptive window, in contrast to the *hSHC* which used a fixed size-sliding temporal window that as our results demonstrated, was less suitable for presenting real-time results. The main research contribution from this study has been the development of an efficient software-only-based sensor calibration algorithm that can improve accelerometer precision, together with the design of a posture-based adaptive signal segmentation algorithm that cooperated with an adaptive hybrid classifier to improve the performance of real-time posture recognition.

**Keywords** Accelerometers · Posture recognition · Adaptive signal segmentation · Real-time classification

## 1 Introduction

Research on human posture recognition has made great progress in recent years having been applied to a wide range of applications, for example surveillance, healthcare, robotics, smart homes, smart cities and human computer interaction [1–4]. Posture recognition aims to automatically classify the physical status of the subject, so as to determine whether the user requires assistance or guidance whenever an abnormal activity is detected. To date, most, posture recognition systems operate by collecting sequential data obtained from cameras or wearable inertial sensors.

In a crowd environment, most existing human activity detection systems have been based on vision technologies [5]. Although vision-based systems have advantages (e.g. are easy to operate and don't require collaboration with those being monitored), they incur various challenges such as the camera position, background clutter, limited coverage, etc. Wearable sensor technologies provide a way of overcoming some of these shortcomings while enabling long-term recording. Wearable inertial sensors are often embedded into various wearable items or devices such as wristbands, shoes, smart phones, smart watches, and clothes [6, 7]. Wearable inertial sensors have a long history going back to the advent of microprocessors where, for example, they were used to create a guidance systems for poorly sighted people [8] to more recent times where advances in technology has made them more feasible and culturally acceptable, since they are unobtrusive, light-weight, low-cost and power-efficient mobile devices [9]. However, they have their own imitations, such as sensor offset, drift, and sensitivity to location on the body.

✉ Shumei Zhang  
zhang-s2@email.ulster.ac.uk

Victor Callaghan  
vic@essex.ac.uk

<sup>1</sup> Hebei IoT Application Technology Innovation Centre, Shijiazhuang University, Shijiazhuang, China

<sup>2</sup> School of Computer Science and Electronic Engineering, Essex University, Colchester, UK

This study was aimed at designing and developing a real-time monitoring system that was suitable for potential security or health incidents. For example, it might be worn by hospital patients, care home residents, police personnel, night-club bouncers or event-stewards (e.g. sports or concert events at a stadium) to summon help should they fall or be knocked to the ground. In order to explore the feasibility of developing such a system, this work sought to identify suitable data pre-processing techniques and model selection approaches. Thus, the main contributions of the paper are:

- The design of an efficient accelerometer calibration algorithm for improving the sensor precision, and reducing its offset and drift.
- The design of a reliable adaptive signal segmentation (ASS) algorithm for posture-based adaptive boundary point detection.
- The development of an adaptive hybrid classifier which combines the above ASS algorithm with an MLP classifier incorporating a plurality voting approach which was evaluated using a real-time posture recognition framework based on simulated crowd security incident scenarios.

The rest of the paper is organized as follows. Section 2 reviews literature relating to data pre-processing and posture recognition algorithms. Section 3 explains the methodologies adopted for sensor calibration and hybrid model design in our prototype, along with an overview of the system architecture. Section 4 describes the experimental setup, procedures and results. Section 5 contains the conclusions and our thoughts on future research directions for this work.

## 2 Related work

Researchers have proposed various daily activity recognition systems using different methods and based on different sensors. For example, Cheok et al. [10] has provided a review of hand-gesture recognition techniques that split the process into different stages: data acquisition, pre-processing, feature extraction, signal segmentation and classification.

Data that is acquired from sensors usually contains errors which arise from various sources including, for example, an improper zero reference. Therefore it is essential to pre-process the data (i.e. sensor calibration), by manufacturer or user, to negate the effects of these errors. Bonnet et al. [11] proposed a unified calibration framework to determine inertial sensor calibration parameters such as sensor sensitivities, offsets, misalignment angles, and mounting frame rotation matrix. Feature extraction and feature selection aims to maximize the classification accuracy while minimize the number of features [12]. In this paper we proposed a new

and efficient sensor calibration method that is performed based-only on six stationary positions.

The purpose of signal segmentation is to divide a signal into several epochs with the same statistical characteristics such as amplitude or frequency. Vari [13] introduced a Modified Varri method for signal segmentation, which includes three parameters affecting the accuracy. These parameters must be determined experimentally, meaning they may not be optimal for any arbitrary signal segmentation application. Azami et al. [14] designed a genetic algorithm (GA) as a powerful search tool to look for appropriate parameters based on the Modified Varri approach. Later, Azami et al. [15] proposed an approach where the signal was initially filtered by a Moving Average or a Savitzky-Golar filter to reduce short-term signal noise aimed at improving the reliability of the method. Novosadová et al. [16] described a polynomial model for signal segmentation that assumes every segment is a polynomial of certain degree. Segment borders correspond to positions in the signal where the model changes its polynomial representation. They also demonstrated that using orthogonal polynomials, instead of the standard polynomial in the model, is beneficial when the segmenting signals are corrupted by noise. Laguna et al. [17] demonstrated a dynamic window method based on events for the signal segmentation. Their approach adjusts dynamically the window size and the shift for each step. This means a new window is generated only when a new event is detected in contrast to the fixed-size sliding window approach. In this paper, we have proposed a hybrid ASS algorithm that combines both the adaptive window and the bottom-up methods for improving the segmentation accuracy.

Human activity classification accuracy is determined by a number of factors such as optimal classifier selection, the data sensing approach, and the data sensing frequency. The work of Gao et al. [18] investigated how the sampling frequency impacted the performance of classifiers, by increasing the sampling rate from 10 to 200 Hz (in 10 Hz increments). They demonstrated that the recognition accuracy was not sensitive to the sampling rate (only a 1% increase above 20 Hz which stabilized beyond 50 Hz). However, the high sampling rate could lead to greater computing load and power requirements. Saini et al. [19] presented a two-person interaction monitoring framework for analysing individual activities in people who may be suffering from some form of psychological disorder, based on the Kinect sensor. They used BLSTM-NN classifier to recognize each individual's activities, and applied a lexicon approach to improve the performance. Their testing achieved a maximum accuracy of 70.72% for 24 different activities. Hegde et al. [20] compared insole-worn and wrist-worn sensors used for daily activities classification, and demonstrated that the recognition accuracy was 81% for an insole-worn sensor alone, 69% for the wrist-worn sensor alone, and 89% for the two

sensors combined. Thus, insole-worn sensors present a compelling alternative or companion to wrist-worn devices. Macron et al. [21] proposed a human gesture recognition system based on volumetric data sequences, and used the HMMs classifier to identify a set of key postures, classifying their sequences over a set of possible actions. They also presented a simple method to identify the number of hidden states of the HMMs for improving the gesture classification performance. Finally, they achieved a 96% accuracy for ten different actions. Naveed et al. [22] introduced methodologies of heterogeneous features fusion for improving the performance of human activity recognition. They employed the time efficiency and optimality of SMO to train an SVM that was tested by a single person and multi-human dataset, achieving an accuracy of 91.99% and 86.48% respectively.

Our review of related work has demonstrated that each algorithm has its advantages and drawbacks. No one classifier works best for every problem as there are several practical factors to consider such as the size and structure of the dataset. Thus, for our machine learning work, it was necessary to employ many different algorithms in order to identify the most appropriate classifier. Hence, for this study, we designed a hybrid classifier (*hAHC*) for real-time posture recognition, which utilizes different algorithms for different tasks.

### 3 Methodologies

Our experimental data collection platform was constructed around ultra-wide band (UWB) anchors and tags. Anchors are fixed UWB hardware nodes, containing at least one so-called Master Anchor responsible for collecting all the data from the other anchors. Anchors send/receive messages to/from mobile tags.

#### 3.1 The system infrastructure

The infrastructure consisted of a number of components as illustrated in Fig. 1.

The sensing and communication hardware comprised a set of openRTLS UWB master anchors, with UWB tags [23], which were used for data collection in this study.

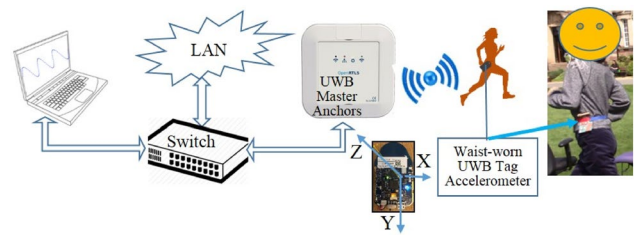


Fig. 1 The system configuration and the tag coordinate system

The tag was housed in a small belt-worn bag on the subjects’ waist, as shown on the right side of Fig. 1. Each tag contained several sensors and actuators such as light, IR proximity, pressure, temperature, and a 9-axes inertial measurement unit (IMU) that included accelerometers (ACCs), gyroscopes (gyros) and, a magnetometer.

The tag communicated with the master anchor wirelessly using a UWB radio. The real-time posture recognition system (ARS) was executed on a laptop which received the tag datasets from the master anchor via UDP messages. Based on our previous work [24], we had determined that the sub-feature set ( $A_x, A_y, A_z, A_{xyz}, \Delta A$ ) had better performance and fewer features compared to other sub-feature sets such as a combination of acceleration data along with associated gyros angles hence, for this study only, the ARS was based on the ACCs datasheet.

#### 3.2 Tri-axial ACCs calibration

The IMU, in our tags, had not been factory-calibrated. The raw measurements ACCs were not specified in  $\pm g$  ( $g = 9.8067 \text{ m/s}^2$ ) level. As a consequence, it was necessary to perform ACCs calibration before the IMU could be used so as to estimate misalignment errors, scale factors, and offsets [25].

The ACCs calibration utilizes the fact that ACCs are affected by gravity when they are under static conditions. The calibrated acceleration  $A_i$  ( $i = x, y, z$ ) and the ACC raw measurements  $\hat{A}_i$  ( $i = x, y, z$ ) can be expressed as in (1).

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = [A_m]_{3 \times 3} \begin{bmatrix} A_{scX} & 0 & 0 \\ 0 & A_{scY} & 0 \\ 0 & 0 & A_{scZ} \end{bmatrix} \begin{bmatrix} \hat{A}_x - b_x \\ \hat{A}_y - b_y \\ \hat{A}_z - b_z \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix} \begin{bmatrix} \hat{A}_x \\ \hat{A}_y \\ \hat{A}_z \end{bmatrix} + \begin{bmatrix} X_{10} \\ X_{20} \\ X_{30} \end{bmatrix} \tag{1}$$

where  $A_m$  is the 3 by 3 misalignment matrix between the non-orthogonal device body axes and the orthogonal ACCs sensing axes;  $A_{sc}$  is the scale factor, and  $b$  is the offset.

$$[A_x \ A_y \ A_z] = [\hat{A}_x \ \hat{A}_y \ \hat{A}_z \ 1] \begin{bmatrix} X_{11} & X_{21} & X_{31} \\ X_{12} & X_{22} & X_{32} \\ X_{13} & X_{23} & X_{33} \\ X_{10} & X_{20} & X_{30} \end{bmatrix} \quad (2)$$

The goal of ACCs calibration is to resolve 12 parameters from  $X_{10}$  to  $X_{33}$ . Thus, with any given raw measurements  $\hat{A}_i$ , the  $A_i$  can be calculated based on (1). In order to obtain the 12 parameters, Eq. (1) can be rewritten equally as (2).

Based on (2), calibration can be performed at 6 stationary positions that include two directions, up and down, for each of the three axes X, Y and Z, based on the tag coordinate system. In theory, an accelerometer will measure the vertical-axis value in  $\pm 1$  g, and along the other two axes, with values of 0 if the unit remains stationary relative to the earth’s surface. Hence, there are 6 ideal outputs  $A_g$  known as  $(\pm 1$  g, 0, 0);  $(0, \pm 1$  g, 0);  $(0, 0, \pm 1$  g) for the 6 static conditions, shown in (3).

$$A_g = \begin{bmatrix} \pm 1g & 0 & 0 \\ 0 & \pm 1g & 0 \\ 0 & 0 & \pm 1g \end{bmatrix}_{3 \times 6}, \quad g = 9.8067\text{m/s}^2 \quad (3)$$

Meanwhile, the sensor’s raw data can be collected, for a given period of time (e.g. 10 s), at each of the 6 stationary positions and averaged for each position thereby reducing the influence of random noise. Thus, there are six measured averaging raw datasets obtained which are shown in (4). Finally, the calibration parameters vector  $X$  can be calculated using (5), according to (2), (3) and (4). Here we apply the least square method in (5). After the calibration parameters matrix  $X$  were determined, the ACCs raw data were calibrated using (2).

$$\hat{A} = \begin{bmatrix} \bar{\hat{A}}_{x1} & \bar{\hat{A}}_{y1} & \bar{\hat{A}}_{z1} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \bar{\hat{A}}_{x6} & \bar{\hat{A}}_{y6} & \bar{\hat{A}}_{z6} & 1 \end{bmatrix}_{4 \times 6} \quad (4)$$

$$X = A_g \cdot \hat{A}^{-1} = A_g \cdot \hat{A}^T \cdot [\hat{A}^T \cdot \hat{A}]^{-1} \quad (5)$$

where  $\bar{\hat{A}}_{xi}$  is the average raw acceleration along the X-axis at each of the 6 positions,  $\hat{A}^T$  means matrix transpose, and  $\hat{A}^{-1}$  means matrix inverse.

Compared to existing IMU calibration approaches, our method is efficient and easy to implement. The parameters matrix  $X$  was calculated offline, then each measured raw ACCs  $[\hat{A}_x \ \hat{A}_y \ \hat{A}_z]$  was calibrated using (2) online. An advan-

tage of this approach is that it required only a software-based calibration phase avoiding the need for special mechanical platforms that feature in many other IMU calibration procedures [26, 27]. In addition, there are fewer calibration parameters (12 vs.18) in our method than in other methods, such as in [28]. These differences result in our approach having a lower calibration cost than most other methods.

### 3.3 Data collection and feature extraction

A calibrated raw dataset was organized as  $(t, A_x, A_y, A_z)$  followed by extracting more features using (6) and (7).

$$A_{xyz}(t) = \sqrt{A_x(t)^2 + A_y(t)^2 + A_z(t)^2} \quad (6)$$

$$\Delta A(t) = |A_{xyz}(t) - A_{xyz}(t - 1)| \quad (7)$$

where  $A_{xyz}$  is the three-dimensional acceleration,  $\Delta A$  is the absolute  $A_{xyz}$  change between time points  $t$  and  $(t-1)$ .

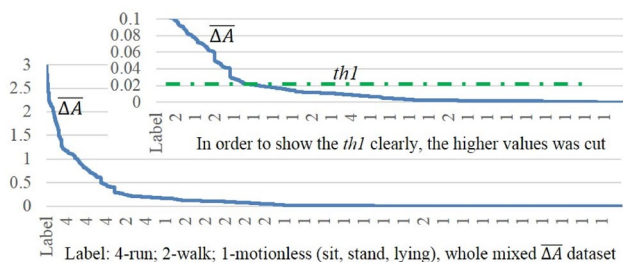
Since the feature  $t$  is only used to record the time points, there are 5 features ( $A_x, A_y, A_z, A_{xyz}, \Delta A$ ) in the dataset used for further processing.

### 3.4 Adaptive signal segmentation (ASS)

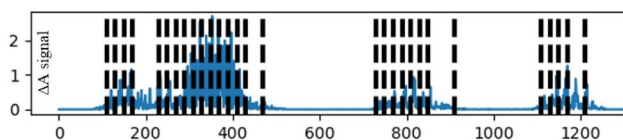
The signal segmentation aims to divide a signal into several periods with similar statistical features. There are two main methods used to do this, static and adaptive segmentation.

The static method divides the signal into fixed periods and is simple and easy to implement. However, it is less reliable, since the duration of the activities are not always equal so, for our work, we chose to use a posture-based adaptive approach to improve the reliability and clarity of the real-time posture recognition. Details of the method are described by the following steps.

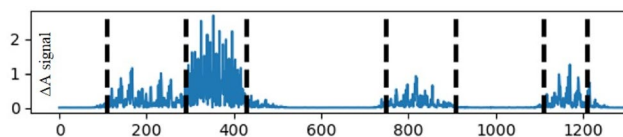
- 1) Define a sliding window size  $w = 2f$  ( $f$  is the sampling frequency) as shown in (8);
- 2) Calculate the difference of average  $\bar{\Delta A}$  between the front half and the rear half sliding window, which is greater than an empirical threshold  $th1$  as shown in (9);
- 3) Calculate the time difference  $\Delta t$  between the middle point of the sliding window and the previous boundary point, which is greater than 2 s as shown in (10).
- 4) Define a boundary point array  $bp[]$ , If the sliding window satisfies both (9) and (10) at the same time, the middle point of the sliding window is determined as the boundary point and saved to the array  $bp[]$ .



**Fig. 2** The experimental threshold  $th1$  based on calculated and mixed  $\Delta A$  dataset (from all subjects). Note: the label on the figure: 4-run; 2-walk; 1-motionless (sit, stand and lying)



**Fig. 3** Based on Eq. (9), the segmentation is correct only for motionless postures. There remain many redundant boundary points for motion postures



**Fig. 4** Based on Eqs. (9) and (10), the segmentation is correct for both motionless and motion postures

$$\begin{aligned}
 & \forall w[i, i + 2f] \in \text{signal} \Delta A \quad (8) \\
 & \overline{\Delta A}[i] = \frac{|\sum_i^{i+f} \Delta A - \sum_{i+2f}^{i+2f+f} \Delta A|}{f} \geq th1 \quad (9) \\
 & \Delta t[i] = t[bp[j]] - t[bp[j - 1]] > 2s \quad (10)
 \end{aligned}$$

where  $th1$  is a threshold defined empirically based on the calculated and mixed dataset ( $\Delta A$ ) as shown in Fig. 2. Here  $th1 = 0.02$ ;  $i = 1, \dots, n$ ,  $n$  is the length of  $\Delta A$  signal;  $j = 1, \dots, m$ ,  $m$  is the length of  $bp[]$  array.

Based on (9), the signal segmentation will be correct only for motionless postures, but there are many redundant

boundary points for motion postures as shown in Fig. 3. Based on (10), the redundant points are eliminated from the motion actions as shown in Fig. 4, since in the short time concerned (less than 2 s) rhythmic posture transition is ignored during a period of the same motion posture (e.g. walking or running period).

Note that for an easy understanding of the procedure, the signal segmentation results shown in Figs. 3 and 4 use the whole signal. However, in the real-time application, the signal was classified every 5 s. This means the signal segmentation was performed for every 100 samples in this study ( $f = 20$  Hz) rather than the whole dataset.

In comparison to existing signal segmentation methods, such as [13, 17], the signals we use ( $\Delta A$  and  $A_{xyz}$ ) are not sensitive to the sensor’s position and orientation, thus making the system configuration more flexible. For example, the tag can be located on the waist, back or pocket, and its direction can be horizontal or vertical, the only restriction being that it remains in the same position for the duration of the session.

### 3.5 Single classifier vs. hybrid classifier

Three classifiers were designed and compared allowing us to determine the best classifier for our system as well as providing a benchmark to wider work. Their details are presented below.

#### 1. Single multi-layer perceptron (MLP) classifier selection

MLP Neural Networks use a gradient descent with back-propagation techniques to learn one or more non-linear hidden layers between the input layer  $X = \{X_i | x_1, x_2, \dots, x_d\}$  and output layer  $Y = \{Y_i | y_1, y_2, \dots, y_n\}$ . In this study, the input  $X = \{X_i | A_x, A_y, A_z, A_{xyz}, \Delta A\}$ . The output defines 5 classes  $Y = \{Y_i | sitting, standing, walking, running, lying\}$ .

A model selection experiment was designed to determine how many hidden layers and how many neurons for each of the hidden layers should be created, where the training and testing dataset were collected separately from the same subject. In doing this a total of 30 models were trained using the same training set. The models varied from 1 to 3 hidden layers, and the number of neurons varied from 6 to 15 at each of the layers. Following this, the 30 models were evaluated

**Table 1** Comparison of classification accuracy for a new dataset, via 30 models with different neuron numbers in each of 3 hidden layers

Neurons ( $n$ )	6	7	8	9	10	11	12	13	14	15
Acc (%)										
MLP( $n$ )	69	67	68	70	70	71	69	61	72	71
MLP( $n, n$ )	71	66	68	70	71	71	<b>73</b>	63	<b>73</b>	<b>73</b>
MLP( $n, n, n$ )	56	58	56	70	68	54	72	67	72	67

Bold values indicate higher accuracy and fewer neurons

using the same testing set. Finally, two hidden layers *single classifier sMLP* (12,12) were selected based on the comparison of experimental results (higher accuracy 73% and fewer neurons) as shown in Table 1, where MLP(n), MLP (n, n) and MLP (n, n, n) represent one, two and three hidden layers respectively.  $n$  is the number of neurons in each of the layers. Here we only utilise a common hidden layer configuration setting which sets the same size for the hidden layers.

During the model training stage, *sMLP* ( $n, n$ ) learns an activation function  $y = f(x, w)$  using (11).

$$y_j = \sum_{k=1}^n w_{kj}^2 \cdot f2 \left( \sum_{k=1}^n w_{kj}^1 \cdot f1 \left( \sum_{i=1}^d w_{ij}^0 \cdot x_i + b0 \right) + b1 \right) + b2 \quad (11)$$

where  $n$  is number of neurons in layer  $l$ ;  $d$  is the dimension of inputs;  $w_{ij}^l$  is the weights, which represents the relationship between two nodes ( $i, j$ );  $b$  is the bias.

The *softmax* function [29] is selected as the activation function ( $f1$  and  $f2$ ), which calculates the probabilities that sample  $x_i$  belongs to each of the classes  $y_j$ . The output is the class with highest probability.

#### 2. Two hybrid classifiers design

The *adaptive hybrid classifier (hAHC)* and a *static hybrid classifier (hSHC)* design are explained in this section.

The *hAHC* combines the adaptive signal segmentation (ASS) (posture-based dynamic window) with the single classifier *sMLP* and plurality voting (PV) approach. The *hSHC* combines the static sliding window (SSW) (fixed-size) with the *sMLP* and PV approach. Here, the SSW uses a 1-s window (with a size of 20 samples, in this study). The three classifiers can be summarized as follows:

*sMLP* = a single MLP (12, 12).

*hAHC* = ASS + *sMLP* + PV.

*hSHC* = SSW + *sMLP* + PV.

The difference between *hAHC* and *hSHC* is their signal segmentation method. Their hybrid implementation procedure is similar and is described below:

- First, the signal was divided into a multiple sub-segments using the ASS or SSW algorithm.
- Then, each of the sub-segments was classified using *sMLP*, with the classification result being saved as an array  $P = [p_1, \dots, p_n]$ , where  $n$  is the sample number within a sub-segment,  $p_i$  is the class label for each sample.
- Finally, for each classification result array,  $P$  was revised using the PV algorithm, which involved counting the number of each class for the array  $P = [p_1, \dots, p_n]$  and obtaining the relevant majority class label ( $maxL[i]$ ). Then each sample within the array  $P$  was set to the same class label, using their  $maxL[i]$ .

Obviously, the *hAHC* and *hSHC* classifiers can reduce the number of possible misclassified samples for each segmented signal. For example, 20 samples of walking action were observed for a sub-segment, out of which 12 samples were classified as walking but 8 samples were classified as standing by the *sMLP*. However, all samples in this segmented signal were labelled as walking by the PV algorithm. Thus, the *hAHC* and *hSHC* classifiers have potential to improve the robustness of the classification process by handling misclassified data.

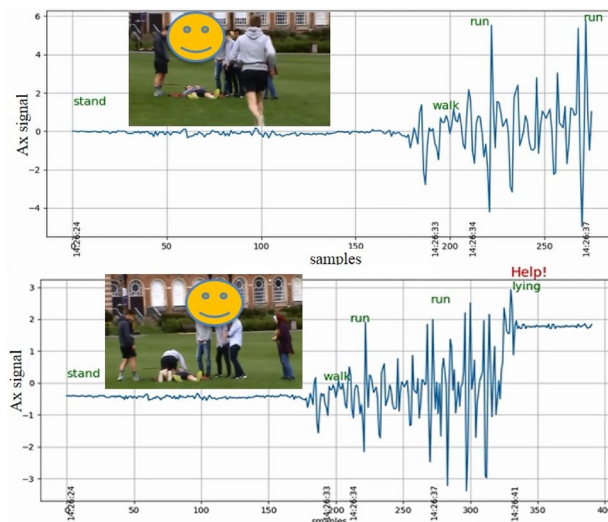
## 4 Experiments

Nineteen subjects attended the experiments. The subjects performed all or part of the predefined 5 test behaviours [*sitting, walking, standing, running, lying*] and data was collected from a waist-worn tag. The experimental results were shown as an animated figure in real-time and validated against synchronized videos that were recorded using a camcorder.

### 4.1 Experimental protocols

The experimental protocols were designed and performed as follows:

- 1) *The training dataset collection*: Six subjects (Sub1-Sub6) performed the 5 actions, described above, in a



**Fig. 5** Screen-shot showing the synchronized video with classified real-time  $Ax$  (collected from the steward) animated figures. The steward's actions were standing, walking, running and lying down. The top and bottom figures show the dynamic real-time posture recognition results

stated order within a laboratory environment. The collected datasets were saved as *trainSet*.

- 2) *The testing dataset collection:* All the 19 subjects were divided into 3 groups. Each group was assigned 1 steward with 5 or 6 visitors, and performed 5 actions randomly based on simulated a security scenario on an outdoor grass field in a public space. Datasets collected from each group were classified in real-time as shown in Fig. 5. Concurrently, the 19 datasets were saved as *testSet* separately.
- 3) *A simulated security scenario:* The 'visitors' playing together on the grass field, a 'steward' was standing or walking. Suddenly, one of the visitors complains he/she had a headache and laid down on the ground, resulting in the remaining visitors summoning the steward with a loud cry of "Help!" In response, the steward ran quickly towards the visitors (the top left of Fig. 5), kneeling (face down lying) to help the unwell visitor (the bottom left of Fig. 5).
- 4) *Models training, testing and comparison:* Three models *sMLP*, *hAHC*, and *hSHC* were trained using the *trainset*, tested using all the 19 *testSet*, and compared using integrated datasets from 13 unfamiliar subjects.

### 4.2 Experimental results and discussion

The experiments were focused mainly on the adaptive signal segmentation (ASS) algorithm evaluation and comparison of the performance among the three classifiers. In addition, we conducted one other experiment to evaluate the signal calibration algorithm.

#### 1. Acceleration calibration algorithm evaluation

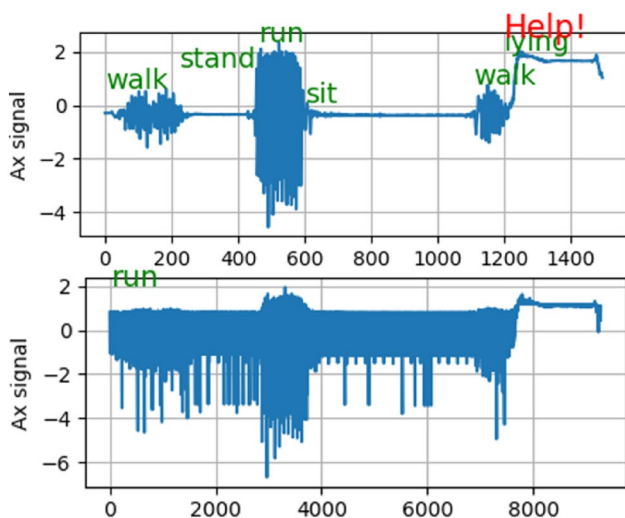


Fig. 6 Classification results based on the calibrated dataset (top of the figure) and the raw dataset (bottom of the figure)

First, one subject performed the 5 specified actions in the order walk-stand-run-sit-walk-lying, while collecting a raw dataset along with the calibrated dataset. Given this sequence of activities, we evaluated the classification performance (using the *hAHC*) by considering the raw and calibrated signal.

The classification results are presented in Fig. 6, which shows that the acceleration calibration algorithm contributed to reducing the measurement noise, thereby improving the classification accuracy. For example, considering the raw dataset, all samples that were erroneously classified as running, could be linked to the large amount of noise present in the sensed data. Additionally, the sample numbers required were reduced by calculating the average value for each 'data sensing' packet during the calibration processing phase. The 'data sensing' packets from the IMU were arranged in the TLV (type-length-value) format. The size of each TLV packet differs, from 6 to 13 values, thus the calibration also contributed to reducing the overall data load.

#### 2. Adaptive signal segmentation algorithm evaluation

The experimental results (see in Fig. 7) indicated that the ASS algorithm was able to obtain the boundary points (bps) correctly for most of the different subjects.

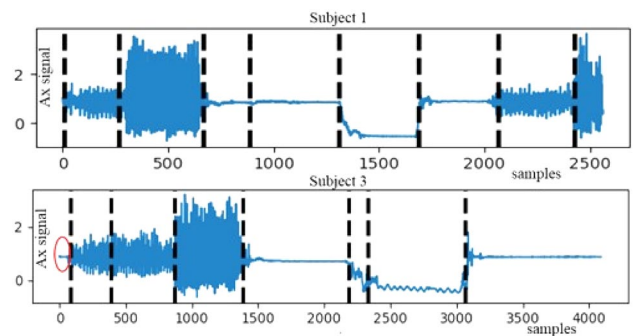


Fig. 7 The ASS algorithm works well for most of the different subjects. Here the Ax signal collected from Sub1 and Sub3 is presented

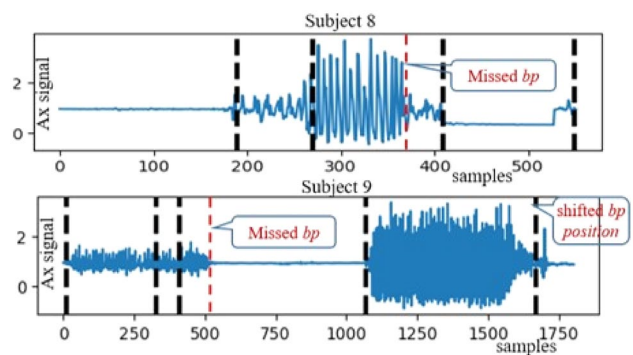
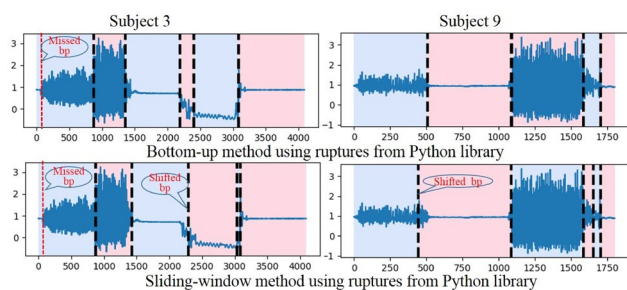


Fig. 8 The ASS algorithm resulted in some incorrect or missed bps for Sub8 and Sub9



**Fig. 9** Signal segmentation results from the bottom-up and sliding-window methods based on signals collected from Sub3 & Sub9

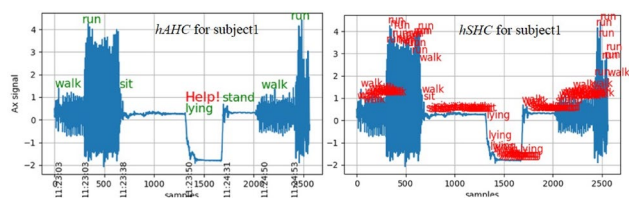
Figure 7 illustrates that the ASS algorithm separated signals for Sub1 and Sub3 correctly. The human test subjects performed the same actions (walking, running, sitting, lying, and standing) for different periods of time, and with their natural (and different) characteristics. Additionally, Sub3 undertook a short period of standing before walking, shown within the red circle at the beginning of Fig. 7.

Figure 8 shows that the ASS algorithm missed some *bps* for signals collected from Sub8 and Sub9, as well as shifting a *bp* position for the Sub9. While the problem can be reduced by increasing the number of human subjects used to learn the parameter  $\theta_1$ , it can never be 100% correct for all subjects, so it needs to operate within an agreed 'acceptable error' for the parameters selection.

Additionally, the proposed ASS algorithm was compared to the bottom-up and sliding-window methods included within the Ruptures package, which is a Python library for offline change point detection [30]. The comparison results, based on signals collected from Sub3 and Sub9, are presented in Fig. 9.

Figure 9 demonstrated that the bottom-up approach had similar performance to the ASS algorithm (the lower graphic in Fig. 7 for Sub3 and Fig. 8 for Sub9). They were both better than the sliding-window method. For example, ASS performed better for Sub3, and the bottom-up approach did better for Sub9, but sliding-window obtained missed *bp* for Sub3 and shifted *bp* for Sub3 and Sub9.

The main advantages of the ASS algorithm were found to be:



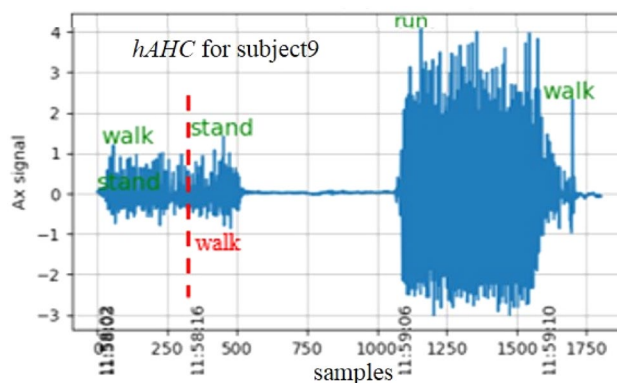
**Fig. 10** Real-time classification results for subject-1 using *hAHC* and *hSHC* classifiers respectively

- The combination of the adaptive window method (divided as motionless/motion windows) and the bottom-up approach (where the redundant points were deleted from the motion windows) improved the segmentation accuracy. This was particularly evident for brief periods of posture, such as the short motionless signal from Sub3 at the beginning of Fig. 7.
- It should also be noted that ASS was ran online, for real-time applications, rather than offline.

### 3. Comparison of the three classifiers' performance

The three classifiers were evaluated in real-time using an animated figure as shown in Fig. 10. This was able to show immediately how many action sections were classified correctly or incorrectly, but could not provide a numerical count, since there were no class labels included in the real-time dataset. Therefore, their performance was compared in two steps: (1) viewing the animated figures in real-time; (2) using classification accuracy gathered from the 'all subjects' offline dataset with manually marked class labels.

Figure 10 illustrated the real-time classification results based on data collected from subject-1. The left side figure was classified using the *hAHC* model, and the right side figure was classified using the *hSHC* model. From these results, it is evident that both classifiers did well. The *hAHC* predicted correct actions for each of the adaptive windows (although some window boundaries shifted slightly). The *hSHC* predicted incorrect actions for only a few sliding windows (e.g. the walking action appeared in the running action period of time). Nevertheless, the *hSHC* showed the results in an overlapping manner that was less clear compared to the results shown via the *hAHC*. In addition, Fig. 10 did not include the result from the *sMLP* classifier, since it shows, for each of the samples, more overlap than the *hSHC* classifier.

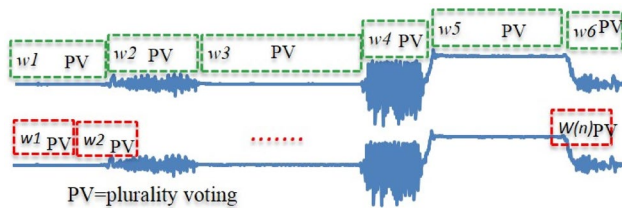


**Fig. 11** Real-time classification result using the *hAHC* classifier for subject-9. Note that part of walking action was recognized as standing since a boundary point was missed by the ASS algorithm.



**Table 2** Comparison of classification reports from the three classifiers evaluated using the same testing set collected from unseen subjects

Classes	<i>sMLP</i>			<i>hAHC</i>			<i>hSHC</i>			Support number
	prec	recall	f1score	prec	recall	f1score	prec	recall	f1score	
Sit	46	84	59	53	91	67	50	84	63	2310
Walk	72	78	75	89	91	90	81	95	87	4629
Stand	71	64	67	81	71	76	78	64	70	3406
Run	92	74	82	94	1	97	99	92	95	3403
Lying	90	60	72	99	61	75	1	60	75	3886
avg/total	76	<b>71</b>	72	86	<b>82</b>	82	84	<b>79</b>	80	17,634



**Fig. 12** Comparison of the difference between the model *hAHC* and *hSHC*. The top figure shows the *hAHC* which used a posture-based adaptive window segmentation with the PV approach; the bottom figure shows the *hSHC* which used a static sliding window segmentation with the PV method

The *hAHC* performance was susceptible to significant influence from the ASS result so, in some cases, the ASS algorithm resulted in a shifted *bp*, or missed some *bp* that will decrease the classification accuracy dramatically as shown in Fig. 11, where part of the walking action was classified as standing, since a boundary point was missed by the ASS algorithm, as is shown in the lower part of Fig. 8.

Three classifiers were compared using their classification report based on integrated datasets from 13 unfamiliar human subjects. The experimental results are shown in Table 2 using the 'precision', 'recall', and 'F1-score' for each class.

The 'precision' (also called the positive predictive value), provides a measure of the ability of the classifier not to label as positive, a sample that is negative. The 'recall' (also known as 'sensitivity'), illustrates the ability of the classifier to find all the positive samples. The 'F1 score' is a weighted average of the 'precision' and 'recall', with the best score being 1 and worst 0.

Table 2 indicates that the *hAHC* model improved the average 'recall' slightly more than the *hSHC*, and significantly more than the *sMLP* (*hAHC* 82%; *hSHC* 79%; *sMLP* 71%) when used with the unseen datasets. The reason is that the three classifiers all classified the signal samples using the single model *sMLP*, but both hybrid models *hAHC* and the *hSHC* combined the sliding window segmentation with *sMLP* and PV algorithms to revise the results from the *sMLP* for each of the sliding windows, using the PV approach as shown in Fig. 12.



**Fig. 13** Different people have different walking gaits, which poses a challenge for developing training models

The difference between both is that the *hAHC* used the posture-based adaptive window (top of Fig. 12), in contrast to the *hSHC* which used the 1-s static window (bottom of Fig. 12), with both obtaining a similar classification accuracy, however the *hSHC* had an overlap issue for results displaying in a real-time animated figure. Thus, is why we selected only the adaptive hybrid classifier *hAHC* for our real-time system.

All three classifiers obtained a higher average 'recall' for datasets collected from the training subjects but their average recall was lower for unseen subjects, since different subjects have different behaviour for the same action as shown in Fig. 13. As is evident from these results, it is challenging to train a suitable common model for all subjects.

In terms of how our work has advanced the state of the art, while the deep learning methods proposed in [31, 32], have unified the feature learning and classification into one model as a way of enhancing the performance of posture recognition, the convolutional neural network (CNN) used requires significant data and time for training the model. For example, Yang et al. [31] used 136,869 and 32,466 samples for training and testing respectively, and spent around 1 h for the CNN training, while spending 8 min for testing. In contrast, the *hAHC* model presented in this study provides a more computationally efficient real-time human posture classification method, which displays better performance with unfamiliar subjects.

## 5 Conclusion and future work

This study was motivated by the need to devise better real-time human posture classification methods for monitoring health incidents at home or monitoring security situations in a crowd environment.

Data-driven models are becoming popular for activity recognition, since their built-in error terms quantify the generalization error [33]. However, previous research has used only a single model which has limitations for some situations. Thus, this work has investigated the potential for combining multiple models with the aim of achieving better performance than using any one of them alone.

In order to improve the posture recognition accuracy and the reliability for use with unseen datasets, three types of classifiers (one single model *sMLP* and two hybrid models: *hAHC* & *hSHC*) were designed and evaluated using a dataset collected from unseen subjects. Experimental results demonstrated that the *hAHC* model performed best when compared to others, since it not only obtained the desired accuracy, but also displayed the real-time results clearly using the posture-based adaptive window that avoids the overlapping issue. In this way the work has advanced the state-of-the-art in classification and posture recognition. Thus, from this work, and supported by the literature review findings, the novelty of this study can be summarized as follows:

- The development of a novel software-only-based sensor calibration algorithm that can improve accelerometer precision efficiently compared to earlier hardware-based methods.
- The creation of an innovative hybrid model *hAHC* which combines three complementary classifiers in three layers (*ASS* + *sMLP* + *PV*), that are able to reduce the number of misclassified samples from the *sMLP* classifier by using the *PV* approach within each posture-based adaptive sliding window, that have each been separated via the *ASS* algorithm, as shown in Fig. 12.
- The adoption of a creative visualization method whereby posture classification that was performed in real-time, is visualized clearly in an animated format.

Additionally, like many other systems, the output of the analysis can provide real-time reminders when an abnormal activity occurs (eg. falls).

Finally, by way of future plans, we aim to investigate the development of a sub-function that will automatically train an individual model for each new user as this would provide a very efficient way for improving the classification system's robustness and reliability.

**Acknowledgement** The authors are pleased to acknowledge funding from European Union, H2020 Grant No. 732350 and the IoT

application technology innovation centre of Hebei province, China, Grant No. SG20182058. We wish to convey a special thanks to Prof. Dorothy Monekosso, the H2020 grant holder at Leeds Beckett University, for co-opting the lead author into this project. We also want to express our gratitude to various staff members at Leeds Beckett University for their kind assistance; and IT Services staff for the loan of camcorders used in the video recording of the experiments. Finally, we should not forget the anonymised people who provided the experimental data, thank you all for supporting this research.

## References

1. Liu T, Kong J, Jiang M et al (2019) Collaborative model with adaptive selection scheme for visual tracking. *Int J Mach Learn Cyber* 10:215–228
2. Cai X, Han G, Song X et al (2019) Gait symmetry measurement method based on a single camera. *Int J Mach Learn Cyber* 10:1399–1406
3. Khan G, Samyan S, Khan MUG et al (2020) A survey on analysis of human faces and facial expressions datasets. *Int J Mach Learn Cyber* 11:553–571
4. Cao Y, Xue F, Chi Y et al (2020) Effective spatio-temporal semantic trajectory generation for similar pattern group identification. *Int J Mach Learn Cyber* 11:287–300
5. Mabrouk AB, Zagrouba E (2018) Abnormal behavior recognition for intelligent video surveillance systems: a review. *Expert Syst Appl*. 91(1):480–491
6. Khokhlov I, Rezni, L, Cappos J, Bhaskar R (2018) Design of activity recognition systems with wearable sensors. *IEEE Sensors Applications Symposium (SAS)*, pp. 1–6
7. Zhang S, McCullagh P, Zheng H, Nugent C (2017) Situation awareness inferred from posture transition and location: derived from smartphone and smart home sensors. *IEEE Trans Hum Mach Syst* 47(6):814–821
8. Freeston IL, Callaghan VL, Russel ND (1984). A portable navigation aid for the blind, in *frontiers of engineering and computing in health care*. IEEE Conference on Engineering in Medicine and Biology Society, Los Angeles, CA, pp. 247–249
9. Cornacchia M, Ozcan K, Zheng Y, Velipasalar S (2017) A survey on activity detection and classification using wearable sensors. *IEEE Sens J* 17(2):386–403
10. Cheok MJ, Omar Z, Jaward MH (2019) A review of hand gesture and sign language recognition techniques. *Int J Mach Learn Cyber* 10(9):131–153
11. Bonnet S, Bassompierre C, Godin C, Lesecq S, Barraud A (2009) Calibration methods for inertial and magnetic sensors. *Sens Actuators A* 156(2):302–311
12. Xue B, Zhang M, Browne WN, Yao X (2016) A survey on evolutionary computation approaches to feature selection. *IEEE Trans Evol Comput* 20(4):606–626
13. Vari A (1988), *Digital processing of the EEG in epilepsy*. Licentiate thesis. Tampere University of Technology, pp. 1–97
14. Azami H, Mohammadi K, Hassanpour H (2011) An improved signal segmentation method using genetic algorithm. *Int J Comput Appl* 29(8):5–9
15. Azami H, Mohammadi K, Bozorgtabar B (2012) An improved signal segmentation using moving average and Savitzky-Golay filter. *J Signal Inf Process* 3(01):39
16. Novosadová M, Rajmic P, Šorel M (2019) Orthogonality is superiority in piecewise-polynomial signal segmentation and denoising. *EURASIP J Adv Signal Process* 6(2019):1–22
17. Laguna JO, Olaya AG, Borrajo D (2011) A dynamic sliding window approach for activity recognition. In: *International conference on user modeling, adaptation, and personalization*, pp. 219–230

18. Gao L, Bourke AK, Nelson J (2014) Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems. *Med Eng Phys* 36(6):779–785
19. Saini R, Kumar P, Kaur B et al (2019) Kinect sensor-based interaction monitoring system using the BLSTM neural network in healthcare. *Int J Mach Learn Cyber* 10(9):2529–2540
20. Hegde N, Bries M, Swibas T, Melanson E, Sazonov E (2017) Automatic recognition of activities of daily living utilizing insole based and wrist worn wearable sensors. *IEEE J Biomed Health Inf*
21. Marcon M, Paracchini MBM, Tubaro S (2019) A framework for interpreting, modeling and recognizing human body gestures through 3D eigenpostures. *Int J Mach Learn Cyber* 10:1205–1226
22. Naveed H, Khan G, Khan AU, Siddiqi A, Khan MUG (2019) Human activity recognition using mixture of heterogeneous features and sequential minimal optimization. *Int J Mach Learn Cyber* 10:2329–2340
23. <https://openrpls.com/shop>. Accessed May 2018
24. Zhang S, Monekosso D, Remagnino P (2018) Data pre-processing and model selection strategies for human posture recognition. *IEEE international symposium on communication systems, networks & digital signal processing*, pp. 1–6
25. Rohac J, Sipos M, Simanek J (2015) Calibration of low-cost tri-axial inertial sensors. *IEEE Instrum Meas Mag* 18(6):32–38
26. Christian JA, Bittner DE, West Virginia University (2018) Apparatus for three-axis IMU calibration with a single-axis rate table. U.S. Patent 9,970,781
27. Hall JJ, Williams RL (2000) Case study: inertial measurement unit calibration platform. *J Robot Syst* 17(11):623–632
28. Harris C, Mahajan S, Leslie P, Spani C et al (2018) Wireless inertial measurement system provides accurate dynamic angle change and angular data for body mount applications. *CMBES Proc* 33(1)
29. Dunne RA, Campbell NA (1997) On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function. In *Proceedings of 8th Aust. Conference on the Neural Networks*, Melbourne, vol 181, p. 185
30. Truong C, Oudre L, Vayatis N (2018) Ruptures: change point detection in Python. arXiv preprint [arXiv:1801.00826](https://arxiv.org/abs/1801.00826)
31. Yang J, Nguyen MN, San PP, Li XL, Krishnaswamy S (2015) Deep convolutional neural networks on multichannel time series for human activity recognition. *International Joint Conference on Artificial Intelligence*
32. Zebin T, Scully PJ, Ozanyan KB (2016) Human activity recognition with inertial sensors using a deep learning approach. In *2016 IEEE SENSORS*, pp. 1–3
33. Luca O (2020) *Model selection and error estimation in a nutshell*. Springer International Publishing, Berlin