



Identifying the walking patterns of visually impaired people by extending white cane with smartphone sensors

Izaz Khan¹ · Shah Khusro¹ · Irfan Ullah²

Received: 16 February 2022 / Revised: 18 January 2023 / Accepted: 21 January 2023 /

Published online: 10 February 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

The loss of or impairment in vision makes it challenging for blind and visually impaired people (BVIP) to navigate easily in their surroundings. Several solutions were proposed to address this challenge and assist BVIP in navigation by exploiting existing technologies. However, their reliance on pre-installed infrastructure and costly dedicated hardware made them less practical. As an alternative, pedestrian dead reckoning techniques were proposed. However, the slow walking pace of BVIP, the required contact with un-intended obstacles, and the false recognition of activities increase error accumulation, making these techniques less applicable. Therefore, solutions are needed to accurately recognize the walking patterns of BVIP so that efficient navigation solutions can be developed. This article fills this research gap by extending traditional white cane with smartphone sensors. Specifically, a smartphone is used with a conventional white cane to collect data through its sensors on a time-based data window. For smooth recording, a revolving tire is attached at the bottom of the white cane. The collected data is processed by employing the computational resources of the smartphone using our designed app, which identifies the user's walking patterns such as walking, stairs up/down, sit/stand, and collision. As a case study, these activities were classified using Naïve Bayes, Random Forests, J48, Decision Table, and LibSVM. Among these, Random Forests gave a higher accuracy. These results suggest that the proposed solution is more practical in designing navigation applications for BVIP and may yield better accuracy if tested with more advanced classifiers.

✉ Irfan Ullah
irfan@sbbu.edu.pk

Izaz Khan
izazcs@uop.edu.pk

Shah Khusro
khusro@uop.edu.pk

¹ Department of Computer Science, University of Peshawar, Peshawar 25120, Pakistan

² Department of Computer Science, Shaheed Benazir Bhutto University, Sheringal 18050, Pakistan

Keywords Blind & visually impaired people · White cane · Smartphones · Assistive technologies · Ubiquitous computing · Human-computer interaction

1 Introduction

Navigation plays a vital role in performing various daily life activities, including going for education, work, shopping, etc., where the critical part of the vision is clear to and acknowledged by every individual [24]. However, the impairment in vision makes it challenging for blind and visually impaired people (BVIP) to navigate easily in their surroundings [21]. Worldwide, at least 2.2 billion people suffer from a near or distance vision impairment [5], a statistic that nobody can ignore. To address this challenge and assist BVIP in navigation, several solutions were proposed by exploiting existing technologies. Numerous navigational tools and apps focused mainly on making technologies navigation-friendly for BVIP. However, these solutions rely heavily on pre-installed infrastructures, such as ultra-wideband (UWB)-based positioning with pedestrian dead reckoning (PDR) [25], as well as dedicated and costly hardware, including, e.g., wearable systems having a depth camera and an embedded computer with haptic feedback for navigation [40]. This reliance makes these solutions less practical to apply in real-world scenarios.

One of the possible solutions to address these issues is to employ PDR techniques [14, 21, 41]. These techniques calculate the current location of a blind user by using her previously known position and advancing it over time using estimated speeds and trajectories (i.e., directions and stride lengths) [14]. These techniques are coupled with building mapped data to improve the confidence of BVIP and assist them more accurately in developing their mental models regarding their surroundings. The use of PDR was a success to some extent, yet the slow walking pace of BVIP and their unattended contact with coming obstacles need special arrangements [22]. Also, it is further affected by the error accumulation due to issues in the carrying position of the inertial measurement unit (IMU), detecting accurate step length, and false activity recognition [23]. Therefore, solutions are required to improve PDR techniques for enhanced navigation. The first step in this regard is to accurately identify the walking patterns of BVIP so that more reliable navigation solutions, tools, and apps can be produced on top of this recognition.

This article fills this research gap by extending traditional white cane with smartphone sensors. The proposed work is inspired by a recent research study [19] that stresses the use of smartphones in developing low-cost and user-friendly navigation solutions. Specifically, a smartphone is used with a traditional white cane to collect data through its sensors and process using its computational resources. The following are the main contributions of this work:

- We use a smartphone with a traditional white cane to collect data through its inertial sensors on a time-based data window.
- We use a revolving tire at the bottom of the white cane for smooth data recording and reducing noise.
- We process the collected data using the computational resources of the smartphone.
- We design a smartphone app to identify the user's walking patterns such as walking, stairs up/down, sit/stand, and collision.
- We perform a case study by experimenting with Naïve Bayes, Random Forests, J48, Decision Table, and LibSVM to classify the walking patterns.

To the best of our knowledge, the work is both novel and original in the sense that no one before has attempted the proposed approach. The novelty lies in the use of smartphone inertial sensors in collecting sensory data, using the computational resources of smartphone for data processing and activity recognition, the development of our app for activity recognition, and extending white cane with revolving tire and keeping the smartphone attached to it for smooth data collection. The proposed work is significant from the perspective of improving the navigation of BVIP. Accurately, recognizing walking patterns will enable the development of more efficient navigation solutions. The resulting higher accuracy, even using some of the basic classifiers, suggests that the proposed solution is more practical in designing navigation applications for BVIP and may yield better accuracy if tested with more advanced classifiers. The rest of the paper spans four sections. Section 2 presents related works and concludes the need to develop the proposed solution. Section 3 gives details about the methodology. Section 4 presents results & discussion. Section 5 concludes the paper, followed by references.

2 Related works

Several technologies have been used to develop assistive solutions for BVIP to help them walk in and navigate their surroundings [34]. These include ultrasonic sensors [29, 44], radio-frequency identification (RFID) tags [3, 23] vision using camera or lasers [10, 39], smartphone technology [20, 21], and sensor data fusion [25, 42]. Among these, several studies have observed smartphones to give orientation and navigation aids to BVIP. The following paragraphs briefly discuss the most related works to draw conclusions and set the stage for presenting the proposed solution.

Vera, Zenteno, and Salas [39] proposed a smartphone-based virtual white cane by combining a smartphone with a laser pointer through a mounting system to detect obstacles in front of the walking user. However, this solution is expensive, consumes too much battery power, and its performance gets affected by the light coming in front of it from other sources. Lin, Lee, and Chiang [27] propose a smartphone-based guiding system for visually impaired users to help them avoid obstacles in their navigation pathway and guide them about their surroundings. It uses a smartphone camera and employs image recognition. It captures the photo of incoming object using the smartphone and sends it for processing to the backend server, which processes it using CNN and YOLO and identifies obstacles with 60% accuracy. However, it is unable to identify and classify activities as walking, stairs down, stairs up, sit/stand, and collision using smartphone sensory data coming from inertial sensors, instead they use camera only for obstacle detection.

Croce et al. [6] developed a smartphone-based indoor navigation system that uses the camera and colored taps stuck to the floor for navigation. The quick response codes (QR codes) are placed on the point of interest, holding the information about locations using uniform resource locators (URLs) to be queried from the server. A blind user must have an active Internet connection so that she can fetch the URLs from the server after scanning QR codes for navigation. These settings make this simple solution costly and limited, as an active fast Internet connection may not be possible each time during navigation.

Amari and Wu [35] used PDR through a smartphone to assist BVIP in navigating indoor environments with auditory feedback. They used mapped routes to limit error accumulation. The errors were handled by updating the current position upon the detection of a change in the heading direction. However, it uses an external wireless IMU with the smartphone, which increases system cost and requires a complex recognition mechanism. Lee et al. [25] proposed an indoor positioning

application for blind people using UWB-based positioning with PDR. They replaced step detection and stride length estimation with deep learning (DL)-based speed estimation using the Kalman filter. The fusion of UWB and DL-based speed estimation have improved positioning accuracy. However, UWB makes its implementation costly due to the required dedicated infrastructure.

Riehle et al. [28] exploited smartphone inertial sensors in assisting BVIP regarding their position in indoor environments. The environment was divided into small cells named microcells having two-dimensional coordinates. The regression-based analysis was used to train the multi-layer perceptron for mapping IMU to the coordinates of the vertex of microcells corresponding to the smartphone's position. Ren et al. [33] experimented with a similar approach with the WeAllWalk dataset considering two scenarios, i.e., with and without the building data. They developed a two-stage turn detector with an LSTM-based step counter to reconstruct the path traversed in the first case. The building map data is used from the mentioned dataset in the second case. Mocanu et al. [30] enhanced the traditional white cane by attaching sensors to interpret colors for walking in smart cities, and the user is warned with tactile signals. Husin and Lim [15] used the internet of things and mobile phones to extend the functionality of the white cane using an embedded board system with various sensors. The data from the sensors are interpreted and passed to the smartphone via Bluetooth for global positioning and SMS alerts. All these solutions are limited to dedicated hardware and costly infrastructure such as smart cities. Therefore, alternatives must be found.

Summarizing, several navigation solutions have been proposed in the literature. However, they are limited in accurately guiding BVIP as their navigation is affected by several factors. These include the sensitivity of the subjects (i.e., the how the sensing device is attached to the body of the user), the complexity of the navigation activity, the insufficient dataset for training/classification, and the constraints on the use of energy and computational resources [7, 16]. The recent literature has extensively used smartphone in assisting BVIP in navigation. They offer portability and convenience to BVIP in navigation [24] and have the potential to be used in developing low-cost and user-friendly navigation solutions [19]. In this regard, the current literature lacks in a smartphone-based solution that can collect data smoothly to improve the classification accuracy. One possible solution is to extend the traditional white cane by adding two modifications. First, smartphone can be placed at a fixed position, near the handle of the white cane, so that the issue of subject sensitivity can be addressed. Second, a revolving tire can be attached at its bottom for smooth data collection. The data collected through smartphone sensors can be processed using the computational resources of the smartphone in identifying the walking patterns of the users. The accurate classification of the walking patterns as walking, stairs up/down, sit/stand, and collision will open new avenues of developing more user-friendly navigation solutions. In line with this need, Section 3 presents the proposed solution and all the essential details of the methodology adopted.

3 Materials and methods

Figure 1 shows some additions and extensions that are suggested to be added to traditional white cane to make it ready for use with our designed smartphone-based solution. We call it smart white cane, which can be created easily by using any selfie stick from the market. Keeping smartphone at the position shown has no negative impact on the use of the white cane, such as folding and joining. A revolving tire is attached at the bottom of the white cane for smooth data collection. We use it with our designed smartphone app, which is

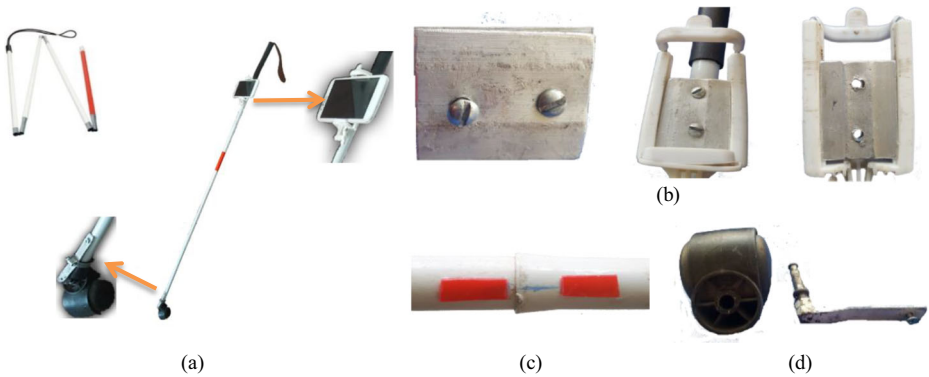


Fig. 1 The white cane after (a) modification, (b) with a holder, (c) joint identification marks, and (d) roller & attachment

schematically presented in Fig. 2, to collect data using sensors, pre-process it, store it, extract features from it, train the classifiers to select a suitable classifier, and use the trained model in identifying walking patterns of the user. The proposed system has been implemented with our designed app to collect sensory data from triaxial accelerometer, gyroscope, and magnetometer from blind users while performing daily activities with the white cane.

The resulting data is used to define a predictive model for identifying walking patterns of BVIP, which include walking, stairs up/down, sit/stand, and collision. This identification can

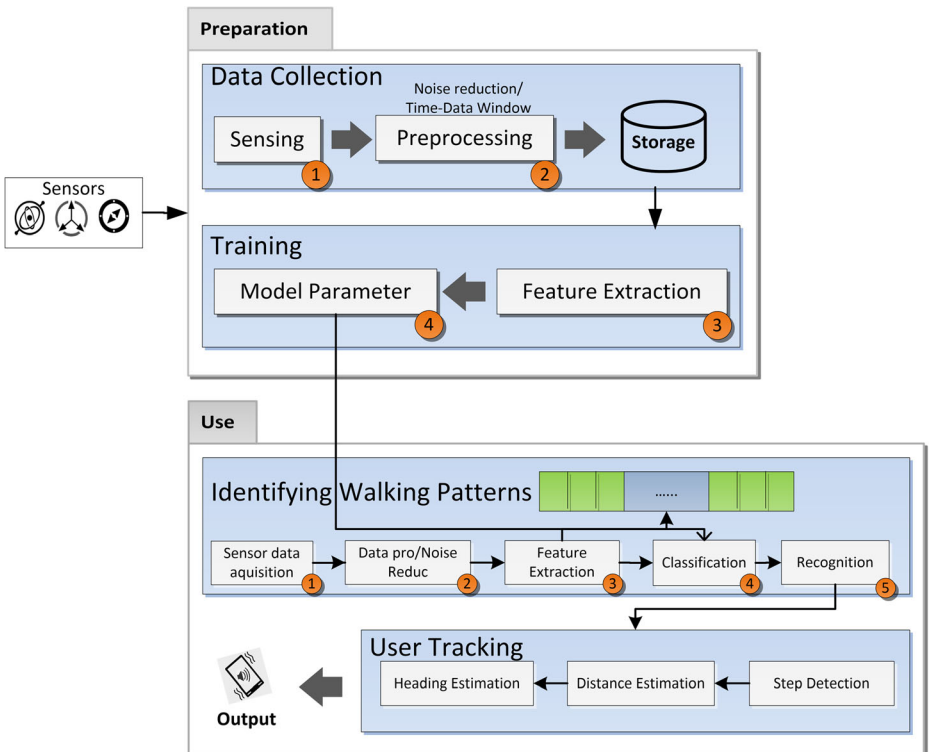


Fig. 2 A schematic diagram of the navigation app for the smart white cane

be done in offline mode using some basic classification algorithms, including Naïve Bayes, Random Forests, J48, etc., on desktop PCs to process raw data and obtain model parameters or online in real-time using smartphone pre-processing & storage capabilities. The collected data is passed through low-pass filters to reduce noise using Android libraries.

The role of feature extraction is inevitable in any inertial sensors-based activity recognition process [17]. The reasons include the limited battery life and the energy-constrained nature of smartphones, requiring to select those features that are energy efficient and give higher accuracy [17]. These features either belong to time or frequency domain [8]. The most widely used time domain features include mean, variance, energy, entropy, correlation between axes, signal magnitude area, tilt angle, and autoregressive coefficients [17]. The most popular frequency domain features include discrete cosine transform and fast fourier transform coefficients [17]. Among these, the latter set of features require long time windows, high computational cost, and are least suitable for real-time applications [17]. Li et al. [26] reported that the time domain features outperform frequency domain features in terms of accuracy for all the reported machine learning algorithms. Therefore, we extracted time-domain features, including mean, median, zero-crossing rate, standard deviation, the correlation between accesses, tilt angle, etc. The resulting data is classified to identify walking patterns using the training data. These activities are exploited in step detection and estimating heading direction and distance based on user profiling to the target position. The following subsections present the essential steps of sensing/data collection, pre-processing & storage, training/classification before they can be exploited in the identification of walking patterns, as presented in Fig. 2.

3.1 Data collection

We developed an Android app to collect data using various smartphone sensors. The smartphone was employed on the said white cane and given to each of the six blind participants

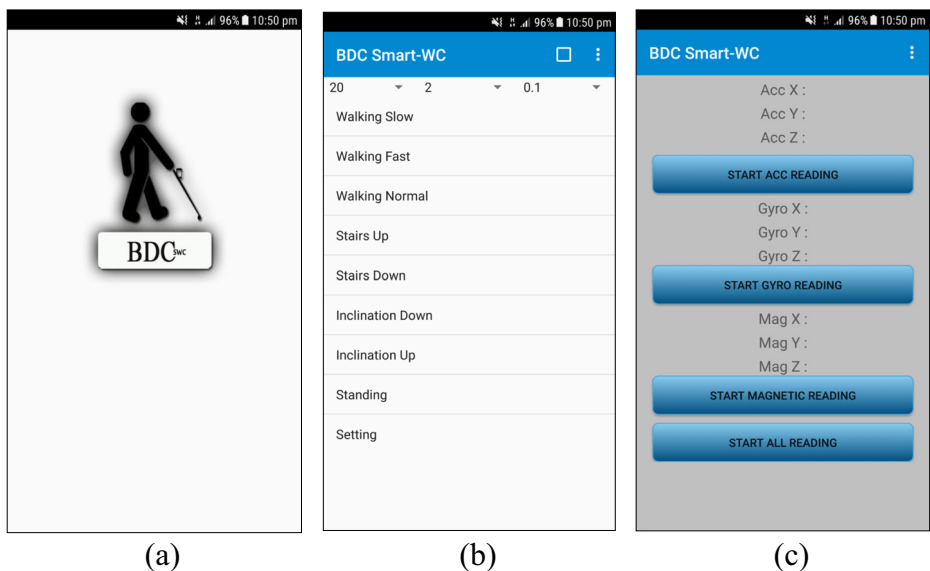


Fig. 3 The Android app for data collection. (a) startup screen, (b) list of activities with soft coded sampling rate, time-data window and noise filter values, and (c) sensor reading with a start and stop button

(male, age: 22–40) from the University of Peshawar, Pakistan. They performed six activities using the proposed smart white cane in an indoor environment. These include walking, stairs up, down, sitting/standing, and collision. Figure 3 shows a few screenshots of the Android app with additional details in Algorithm-I & II and Section 3.3. For the data collection, storage, & training/classification, the proposed app offers all activities with the option for the BVIP to select any activity by dragging her finger up to down. The data collection can be started for a specified activity by double tapping the screen. Algorithm-I presents the programming logic used behind the smartphone app for data collection, pre-processing, and storage.

Algorithm I Data collection, pre-processing, and storage

INPUT: t_w , α , activity name, s_r .

OUTPUT: The string names of the recognized activities.

1. Create list X , Y , Z
 2. Initiate each sensor with s_r
 3. When the user starts the activity
 4. | Call *getSensorsData*(α)
 5. | Call *startTimer*(t_w)
 6. When the user stops the activity
 7. | Release resources
 8. Procedure *getSensorsData* (α)
 9. | Remove gravitational factor using Equation 2
 10. | Remove noise using low pass filters
 11. | Write sensors data into the list X , Y , and Z
 12. Procedure *startTimer*(t_w)
 13. | Create timer
 14. | Run timer
 15. | | Pass the list data into *FeatureExtractor* (list X , Y and Z)
 16. | | Clear list X , Y and Z
 17. | Repeat timer after t_w
-

3.2 Data pre-processing and storage

The collected data is pre-processed to reduce noise, create a time-based data window, and store in a format suitable for further analysis. Two types of noise comes with the data collection, the one generated by sensors when an activity is started or stopped by the user and the second from accelerometer sensor that registers two forces along the X , Y , and Z dimensions. The former force is applied by the user on the smartphone [37]. The latter is constant gravitational force g due to earth gravity [37], handled using Eq. 1 and 2. Eq. 1 computes the applied acceleration to the device A_d , measured as the sum of actual forces applied to the sensor itself F_s and the body mass m [37]. Equation 2 computes the actual acceleration displacement by removes g [37]. The value of α is soft-coded to get a value suitable for data filtration. V shows the value of acceleration for particular axis at time t instance.

Table 1 Noise reduction in start/stop of activity with the averaging method

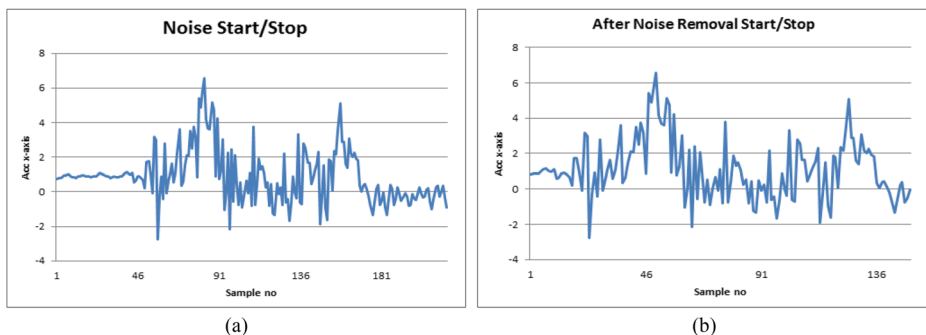
S #	No of readings before the activity start	Time taken before the activity starts	No of readings after activity stops	Time taken after activity stops
1.	42	0.9333	39	0.8667
2.	39	0.8667	42	0.9333
3.	33	0.7333	33	0.7333
4.	31	0.6889	32	0.7111
5.	29	0.6444	41	0.9111
6.	32	0.7111	35	0.7778
Total	206	4.5778	222	4.9333
Average	34	0.7630	37	0.8222

$$Ad = -g - \sum Fs/m \quad (1)$$

$$g = (1-\alpha) * g + \alpha * (Vt-g) \quad (2)$$

In Table 1, we have performed analysis for finding the initial and final desired values of the number of readings for starting time-based data windows and filtration. Figure 4 shows the results before and after removing noise while starting/stopping activities. The low pass filter is used to remove noise, where different α values are used, as shown in Fig. 5. During noise removal, it was observed that if α is higher, the resulting data becomes zero, which is not required as we are interested in noise removal only. Therefore, $\alpha = 0.2$, which removes noise without affecting the actual data.

Dividing data into a time-based data window is among the most suitable techniques for activity recognition [45]. A feature vector (time-domain features) of each time-data window is computed and used in the learning phase. Activities are either performed sequentially or concurrently [1]. The time-based data window is suitable for sequential activities [45]. In our case, the data acquisition is sequential; therefore, a time-based data window approach was used. However, selecting the appropriate window size is challenging, significantly affecting recognition accuracy and system response time. Therefore, different window sizes were chosen based on time [18, 37] and a range of 2–12 seconds. The accuracy was increased by using the sliding window approach [32, 38], which is effective if there are transitions between activities data (reduce transition state noise) [4, 31, 43]. In our case, there is no transition among the selected activities, and therefore, we use the time-based data windows concept only. The raw data for activity recognition is kept in the smartphone's internal memory and

**Fig. 4** The noise due to starting/stopping an activity (a) before and (b) after its removal

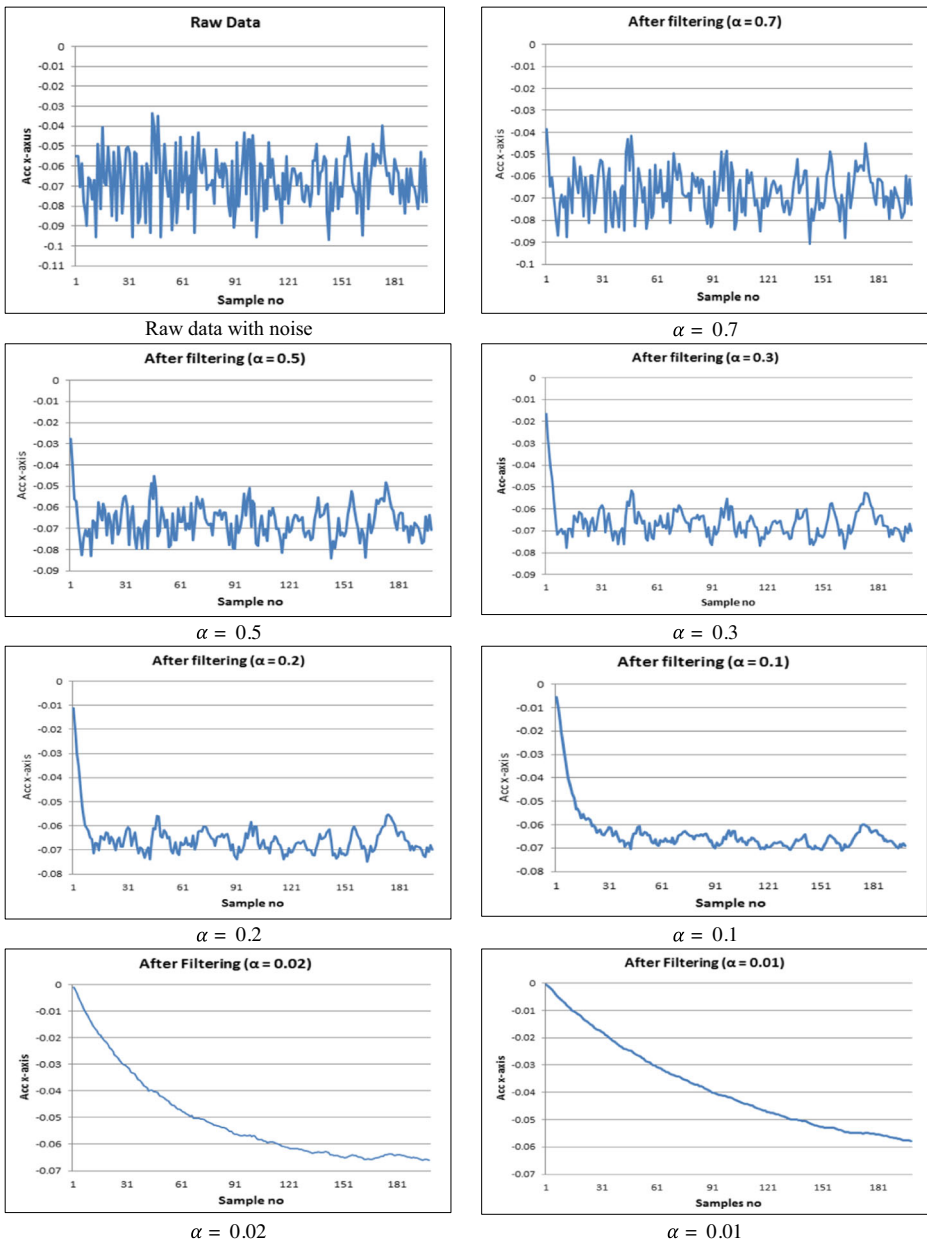


Fig. 5 The best value of α for noise removal

transferred to the computer for analysis. The text file format (.txt) was used for data storage because of its simplicity and easy processing during data analysis. The data recorded by each sensor is stored in a separate text file naming with sensor name date and time, e.g., (ACC: 12 Feb 2017 11:55:36.txt). In the text file, the three-axis data is stored and separated by tab labeling with the desired axis, such as Ax, Ay, and Az, as shown in Fig. 6.

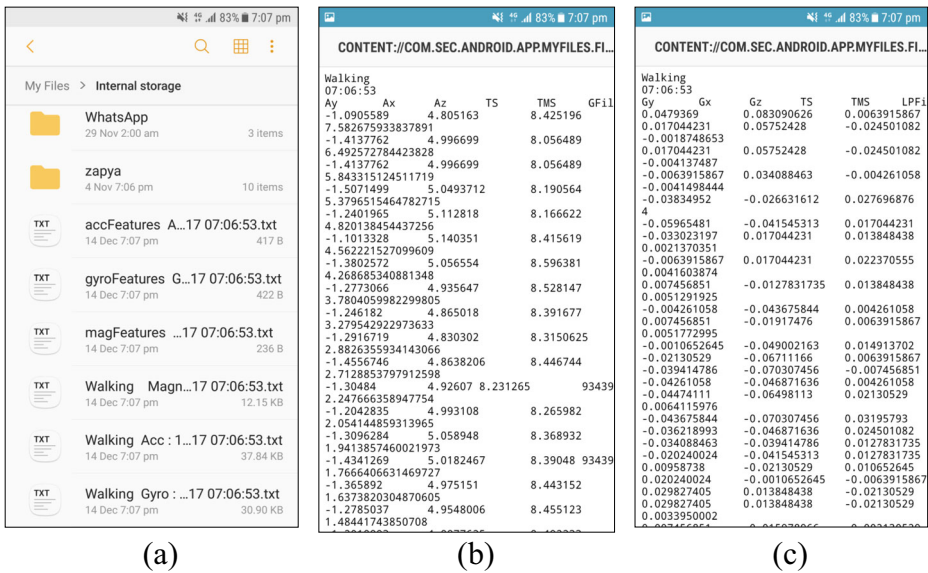


Fig. 6 Activity data storage test file with sensor axis labels

The proposed solution collects sensors' data from various inertial sensors of the smartphone. From the collected data, different features are extracted. As discussed early, extracting and processing frequency domain features consume too much resources of the smartphone [8, 17]; therefore, we extracted time domain features. Specifically, we extracted simple statistical features (SSF), a type of time domain features, which are better for recognizing activities due to low resource consumption [17]. This study used SSF as it consumes less power/energy of the smartphone and offers high accuracy at low sampling rates [12, 18].

3.3 Training

The extracted features help train the classifier to classify the activities in real-time. The Waikato Environment for Knowledge Analysis (WEKA)¹ tool [11] was used for employing the learning algorithms to evaluate a given dataset using random split or cross-validation. WEKA also supports APIs through which new learning algorithms can be incorporated. The time-domain features hold basic statistical data for each data window (segment). These include mean, standard deviation, correlation, zero-crossing rate, and maximum and minimum values. The three-axis sensory data is considered, but some studies consider signal magnitude area as the fourth dimension. It is the sum of the three-axis of the sensor in each segment window and used in cases where orientation sensitivity is considered an issue, unlike other axes of the sensors [18]. In our case, orientation is controlled with a fixed mounting system on the smart white cane that neglects the need for the fourth dimension. The mean is the average value for each time data window for each dimension in the mentioned features. Max and min represent the maximum and minimum reading in each time-data window. The standard deviation represents the amount of dispersion in the dataset related to their mean value, and correlation estimates the strong or weak linkage between the pair of axes. The zero-crossing rate indicates

¹ <https://www.cs.waikato.ac.nz/ml/weka/>

the number of times the value sensed is less than zero in each time-data window. This discussion leads to three axes for each of the sensors. We have six time-domain features for each time-data window: mean, correlation, standard deviation, zero-crossing rate, and min and max values. In total, we have 18 features for each sensor and 54 features for all sensors. The selection and processing of the feature extraction are kept simple to assess the role of sensors and their combination in identifying the walking patterns through this specialized setting. Algorithm I and II show each of the steps performed for the data collection, preprocessing, storage, and feature extraction.

Algorithm II Identify walking patterns

INPUT: t_w , α , activity name, s_r .
OUTPUT: The string names of the recognized activities.

18. Create list X, Y, Z
19. Initiate each sensor with s_r
20. When the user starts the activity
 21. Call *getSensorsData*(α)
 22. Call *startTimer*(t_w)
 23. Call procedure *classifyActivity* (list of features)
24. When the user stops the activity
 25. Release resources
26. Procedure *getSensorsData* (α)
 27. Remove gravitational factor using Equation 2
 28. Remove noise using low pass filters
 29. Write sensors data into the list X, Y, and Z
30. Procedure *startTimer*(t_w)
 31. Create timer
 32. Run timer
 33. Pass the list data into *FeatureExtractor* (list X, Y and Z)
 34. Clear list X, Y and Z
 35. Repeat timer after t_w
36. Procedure *classifyActivity* (list of features)
 37. Apply a suitable classifier to classify a given activity based on the features provided.
 38. return recognized activities

The processed sensory data was labeled with the corresponding activity by the Android app using supervised learning. These activities were classified using Naïve Bayes, Random Forests, J48, Decision Table, and LibSVM. WEKA is used to obtain the model parameters (vector of features, i.e., the trained dataset) and the significant classifier for evaluation and implementation. The extracted features were converted into Attribute-Relation File Format (ARFF) before inputting to WEKA, the required format for this platform. The most optimal classifier was selected for the design of the test application by applying 10-fold cross-

validation on the same dataset after the evaluation. The details are further discussed along with the results in the coming section.

4 Results and discussion

This section attempts to identify the most important factors for identifying the walking patterns of BVIP, such as the sampling rate, the reasonable axis of the sensors to identify accurate activity, and time-based window size. It also attempts to find the most reliable classifier for identifying the walking patterns of BVIP.

4.1 Finding the reliable sensor axis

Each sensor is subject to change concerning data it collects while the user navigates at a given time. Therefore, it becomes active to collect reliable data and represent the actual scenario. As each sensor gives three-dimensional readings along the x-, y-, and z-axis, it is essential to analyze the data for each activity on specified access and find out the reliable axes that could be included in recognizing a specified activity. This requires data sampling for each activity performed by each user holding the smart white cane with a basic usage position to get the initial sensory readings. These initial readings (default values), shown in Fig. 7 for different sensors, help understand the sensor axes and compare various activities. For example, the default values for the accelerometer sensor are approximately 1, 6.4, and 7.2 for the x-, y-, and z-axis, respectively. For each axis of the gyroscope, these values are near to zero.

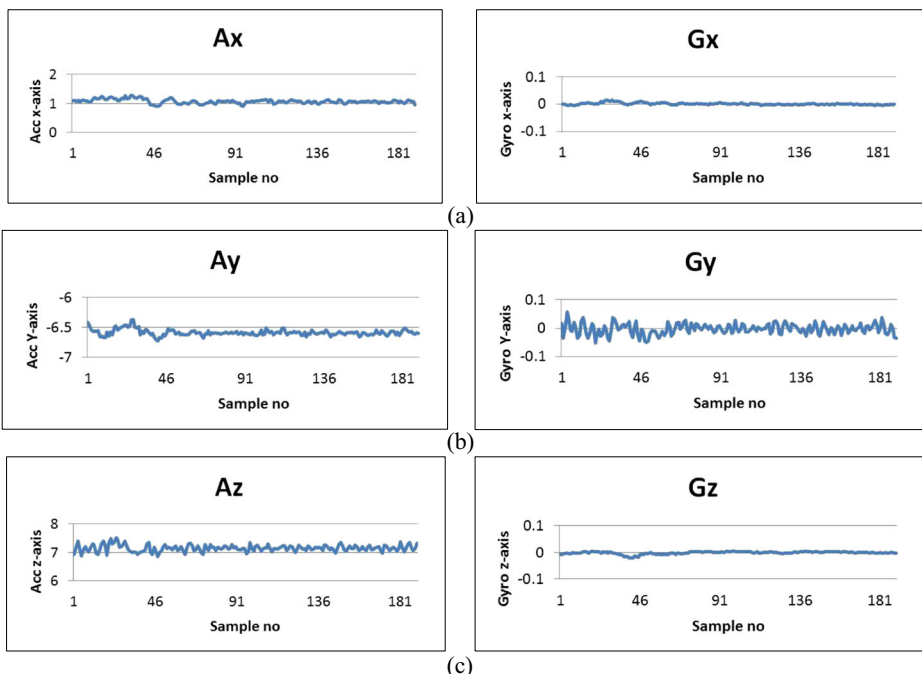


Fig. 7 Default axis of smartphone accelerometer and gyroscope sensors. (a) x-axis, (b) y-axis, and (c) z-axis

The activities performed by blind participants several times to compare the collected data against the given default values for understanding changes in the collected sensory data for each activity. Each activity is graphically illustrated in Fig. 8 and Fig. 9 for accelerometer and gyroscope, respectively. It can be seen that walking is better recognized by the x- and y-axis of the accelerometer. The z-axis partially contributes to this recognition in the maximum and minimum range of (4, -4) for the x-axis and (-4, -12) for the y-axis. However, the gyroscope gives considerable readings for walking in all three dimensions in the range of (0.4, -0.6), (4, -4), (4, -4) for x-, y-, and z-axis, respectively. Therefore, it can be used to identify walking easily. The sensor readings for the remaining patterns are shown in Fig. 8 and Fig. 9. The accelerometer’s x-axis offers partial patterns for stairs-up, while the gyroscope’s x-axis gives reliable patterns. For the stairs down, all axes of the gyroscope and x- and y-axis of the accelerometer participate completely without the z-axis. For standing/sitting, the readings of the accelerometer’s x-axis and gyroscope’s y- and z-axis show no clear patterns and readings vary irregularly, but all other axes are participating well. For collision, all the accelerometer and gyroscope axes participated fully.

4.2 Time-based data window and sampling rate

The appropriate data window and sampling rate for accurately identifying the walking patterns was calculated by observing BVIP in real-time to see how much time they take in navigation. We created two datasets with configurations given in Table 2. Each dataset is arranged based on the sampling rate. We collected data for each activity with basic settings, e.g., 20 samples/sec with different time-based data windows ranging from 2.0 to 3.0. This resulted in two

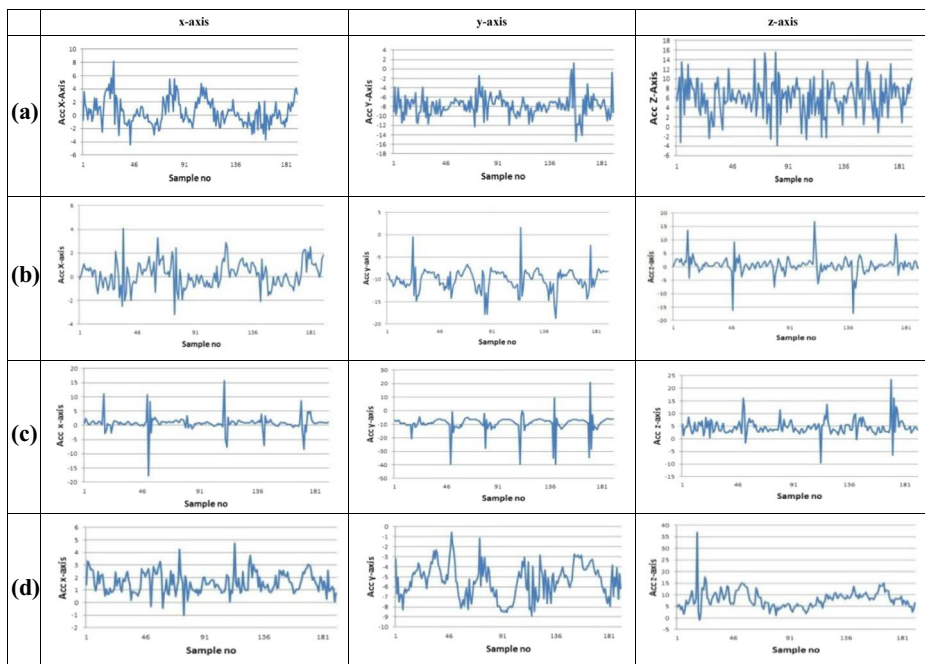


Fig. 8 The data representation of accelerometer sensor for different activities. (a) Walking (b) Stairs Up (c) Stairs Down (d) Sit/Stand

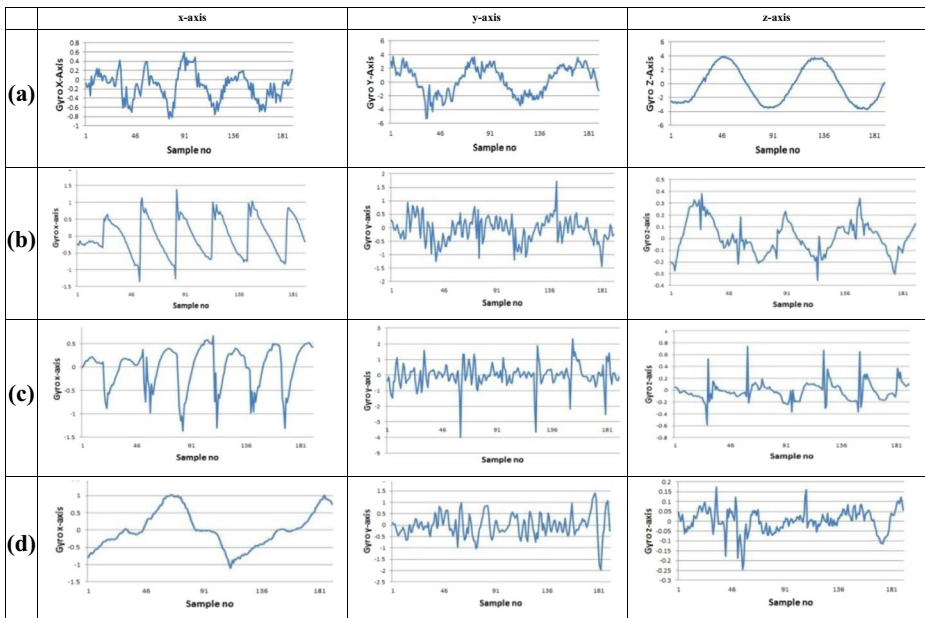


Fig. 9 The data representation of gyroscope sensor for different activities. (a) Walking (b) Stairs Up (c) Stairs Down (d) Sit/Stand

datasets, each having three small datasets with the same sampling rate but different time-based data windows. These datasets gave the evaluation required for developing our design smartphone app for identifying walking patterns. Table 3 summarizes the accuracies of the selected classifiers using these datasets. The most reliable setting for achieving the best results is a sampling rate of 40 with a data window size of 2.0 and 3.0 seconds. However, to guide the blind users in real settings, a shorter time-based data window is required so that the target application can quickly alert the user. Therefore, we considered the sampling rate as 20 and the time-based data window of 2.0 seconds. This setting yielded, on the average, an accuracy of 95.5%, which is sufficient to be used in the smartphone app to identify the walking patterns of BVIP.

4.3 Experimental results in real-settings

We collected data from the accelerometer and gyroscope sensors to perform the experiments in real settings according to the time-based data window. Five parameters were extracted from each sensor’s data with labeled classes based on the time-based data window, resulting in a

Table 2 The datasets time-based data window and sampling rates

Dataset #	Activities	Sampling Rate	Time-based data window
Dataset 1	Walking, stairs up, stairs down, sit/stand, collision	20	2.0
			2.5
			3.0
Dataset 2	Walking, stairs up, stairs down, sit/stand, collision	40	2.0
			2.5
			3.0

Table 3 The classifiers' accuracies based on sampling rates and time-based data windows

Sampling Rate	Time-data window	Activity	The % accuracy of the classifier				
			Naive Bayes	J48	Random Forrest	Decision Table	LibSvm
40	2.0	Walking	98.3	98.3	98.1	96.7	74.2
		Stairs Up	83.3	94.2	97.0	87.8	98.9
		Stairs Down	90.3	94.9	96.1	90.3	99.9
		Sit/Stand	88.9	93.6	97.0	82.4	99.9
		Collision	94.2	95.3	98.0	85.0	83.4
		Total	91.0	95.2	97.2	88.4	91.2
40	2.5	Walking	99.4	99.8	98.8	99.0	74.5
		Stairs Up	59.5	59.1	97.2	71.4	0.00
		Stairs Down	86.2	96.0	96.1	81.8	0.00
		Sit/Stand	98.9	94.0	95.9	95.0	100
		Collision	90.2	88.3	97.3	88.8	88.7
		Total	86.8	87.4	97.0	87.2	52.6
40	3.0	Walking	99.2	99.7	98.6	98.6	71.7
		Stairs Up	86.1	84.4	95.1	76.5	100
		Stairs Down	83.2	88.9	96.9	77.8	0.00
		Sit/Stand	99.3	97.7	97.2	96.4	100
		Collision	92.3	93.0	98.2	90.0	88.0
		Total	92.0	92.7	97.2	87.8	71.9
20	2.0	Walking	95.2	93.2	96.1	95.7	73.2
		Stairs Up	80.1	92.1	95.2	84.3	96.9
		Stairs Down	87.3	91.4	95.1	91.3	94.9
		Sit/Stand	83.2	90.1	95.0	80.1	91.9
		Collision	88.8	92.3	96.1	88.2	81.3
		Total	86.9	91.8	95.5	87.9	87.6
20	2.5	Walking	94.2	94.2	95.8	97.3	72.3
		Stairs Up	53.1	53.2	94.2	69.2	3.00
		Stairs Down	82.4	95.4	94.4	78.3	0.00
		Sit/Stand	95.2	91.1	95.1	93.1	95.5
		Collision	88.2	93.0	96.0	88.3	82.3
		Total	82.6	85.3	95.1	85.2	50.6
20	3.0	Walking	94.4	94.1	98.2	93.6	61.3
		Stairs Up	82.5	81.2	91.3	72.5	77.8
		Stairs Down	97.3	86.1	95.2	71.8	3.00
		Sit/Stand	97.1	93.1	94.2	92.4	99.0
		Collision	93.0	90.0	96.9	91.0	92.3
		Total	92.8	88.9	95.6	84.2	66.8

total of ten parameters for each activity. Table 4 shows the number of times each of the six blind users performed an activity. The collected data was exploited in training and testing. The selected classifiers were employed to classify these activities, where 10-fold cross-validation was used for evaluation, with each fold holding an equal proportion of each class [36]. The equal distribution guarantees the high quality and precision of results without causing any ambiguity [13]. The WEKA version 3.8 was used without applying any optimization on this dataset of 5910 instances with their labeled classes. Each walking activity used ten parameters for real type data and one nominal data type to represent the class.

Table 5 shows the confusion matrix for the walking patterns including walking, stairs down, stairs up, sit/stand, and collision. This confusion matrix [2, 43] is automatically generated by WEKA with computation of all the required parameters, such as true positive (TP) and misclassifications or false positive (FP) required for the evaluation of accuracy and

Table 4 Sample collected by each blind user for different walking patterns

User	Walking	Stairs down	Stairs up	Sit/stand	Collision	Total
1.	275	197	150	162	263	1047
2.	250	180	130	144	280	984
3.	230	170	132	145	300	977
4.	260	180	140	134	230	944
5.	280	188	133	120	250	971
6.	249	210	145	150	233	987
Sum	1544	1125	830	855	1556	5910
%	26.12	19.03	14.04	14.46	26.3	100

computation of precision score, i.e., $Precision = TP/(TP + FP)$ [9]. For example, in Table 5, first row, the bod-faced value reports $TP = 1470$ and normal-faced values represent $FP = 9 + 20 + 45 = 74$. Putting these values in the above formula gives 95.2%, as given in the first row under the precision column in Table 5.

For walking, Random Forests outperforms others by achieving an accuracy of 96.1% followed by Decision Table with 95.7%, Naïve Bays with 95.2%, J48 with 93.2%, and

Table 5 The confusion matrices for different classifiers in identifying the walking patterns

Activity	Classified as					Precision
	Walking	Stairs down	Stairs up	Sit/stand	Collision	
Naive Bayes						
Walking	1470	9	0	20	45	95.2%
Stairs Down	0	982	0	65	78	87.3%
Stairs Up	23	55	674	78	0	80.1%
Sit/Stand	102	0	41	712	0	83.2%
Collision	123	0	0	50	1383	88.8%
J48						
Walking	1440	0	15	29	60	93.2%
Stairs Down	0	1029	34	47	15	91.4%
Stairs Up	6	23	765	12	24	92.1%
Sit/Stand	42	0	3	771	39	90.1%
Collision	55	11	21	32	1437	92.3%
Random Forrest						
Walking	1485	0	10	20	29	96.1%
Stairs Down	0	1070	8	35	12	95.1%
Stairs Up	9	6	790	0	25	95.2%
Sit/Stand	18	20	4	813	0	95.0%
Collision	26	21	13	0	1496	96.1%
Decision table						
Walking	1478	16	0	30	20	95.7%
Stairs Down	48	1028	0	31	18	91.3%
Stairs Up	0	30	700	0	100	84.3%
Sit/Stand	170	0	0	685	0	80.1%
Collision	103	0	80	0	1373	88.2%
LibSVM						
Walking	1131	43	13	36	321	73.2%
Stairs Down	57	1068	0	0	0	94.9%
Stairs Up	25	0	805	0	0	96.9%
Sit/Stand	150	0	0	695	10	81.3%
Collision	180	0	0	12	1364	87.6%

The bold-faced values that appear diagonally represent those activities that were correctly classified, while the remaining ones are misclassifications

LibSVM with 73.2%. For stairs up LibSVM outperforms others by showing an accuracy of 96.9% followed by Random Forests with 95.2, J48 with 92.1%, Decision Table with 84.3%, and Naïve Bayes with 80.1%. For stairs down Random Forests outperforms others by showing an accuracy of 95.1% followed by LibSVM with 94.9%, J48 with 91.4%, Decision Table with 91.3%, and Naïve Bayes with 87.3%. For sit/stand, Random Forests outperforms others by showing an accuracy of 95% followed by J48 with 90.1%, Naïve Bayes with 83.2%, LibSVM with 81.3%, and Decision Table with 80.1%. For collision, Random Forests outperforms others by showing an accuracy of 96.1% followed by J48 with 92.3%, Naïve Bayes with 88.8%, Decision Table 88.2%, and LibSVM with 87.6%. It is worth noting that the WEKA software enables handling the class/data imbalance by offering cost-sensitive classification, which penalizes those classes that are misclassified. However, in the confusion matrix of Table 5, the majority of the classes are correctly classified, while misclassifications are at a minimum. Another possible reason behind some of the misclassifications is the resemblance or similarities among activities, such as between walking and collision, which led to misclassifications in some scenarios. It can be seen that for a majority of the walking patterns, including walking, stairs down, sit/stand, and collision, Random Forests outperforms others. Therefore, in the implementation of Algorithm-II in our designed smartphone app, it can be implemented as a suitable classifier. In addition, Random Forests gave fewer overlapping results with values 29, and 26 for walking and collision, respectively, and thus are much lesser than the corresponding values of 60 and 55 in the case of J48. Figure 10 shows the activity recognition application based upon the selected classifier.

Table 6 presents the results obtained from the classification of walking patterns by using the smartphone app with trained classifiers in real-settings. The accuracies of the classifiers are presented with associated activities. It can be seen that all classifiers performed well in classifying the activities accurately. These accuracies resulted in from the device orientation at a fixed position on the white cane, the specified time-based data window, and using the revolving tire for smooth data collection. Moreover, the sensor data fusion of the accelerometer and gyroscope led to the maximum accuracy of classifying the walking patterns.

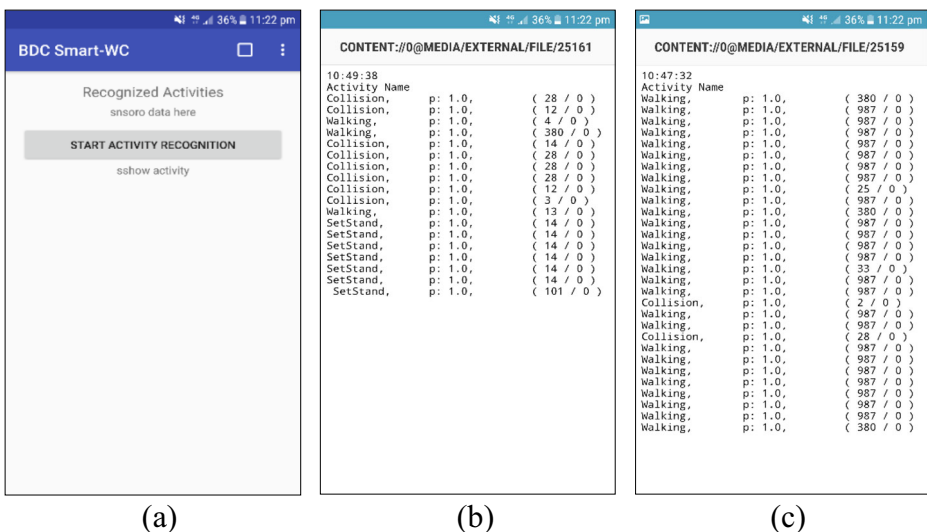


Fig. 10 Activity recognition app based on the selected classifier

Table 6 The percentage accuracies of the classifier to identify walking patterns

Activity	Naive Bayes	J48	Random Forrest	Decision Table	LibSVM
Walking	95.2	93.2	96.1	95.7	73.2
Stairs up	80.1	92.1	95.2	84.3	96.9
Stairs down	87.3	91.4	95.1	91.3	94.9
Sit/stand	83.2	90.1	95.0	80.1	91.9
Collision	88.8	92.3	96.1	88.2	81.3
Average	86.9	91.8	95.5	87.9	87.6

5 Conclusions and future work

The loss of or impairment in vision makes it challenging for blind and visually impaired people to navigate easily in the surroundings. To address this challenge and assist them in navigation, several solutions were proposed by exploiting existing technologies. However, their reliance on pre-installed infrastructure and dedicated costly hardware made them less practical. As an alternative, the pedestrian dead reckoning techniques were proposed. However, the slow walking pace of blind and visually impaired people, the required contact with un-intended obstacles, and falsely recognizing activities increase error accumulation, making these techniques less applicable. This article extended traditional white cane with the smartphone sensors. We used a smartphone with traditional white cane to collect data through its sensors on a time-based data window. For smooth recording, we attached a revolving tire at the bottom of the white cane. The collected data was processed using the computational resources of the smartphone via our designed app, which identifies the walking patterns of the user such as walking, stairs up/down, sit/stand, and collision. As a case study, these activities were classified using several classifiers including Naïve Bayes, Random Forests, J48, Decision Table, and LibSVM. The following are the main findings and conclusions of this study.

- The placement of smartphone at the specified position as shown in Fig. 1 and the use of revolving tire improves the data collection accuracy of the sensors, where even the basic classifiers achieve, on the average, more than 95% accuracy for all the reported walking activities.
- For walking, Random Forests outperforms others by achieving an accuracy of 96.1% followed by Decision Table with 95.7%, Naïve Bays with 95.2%, J48 with 93.2%, and LibSVM with 73.2%.
- For stairs up LibSVM outperforms others by showing an accuracy of 96.9% followed by Random Forests with 95.2, J48 with 92.1%, Decision Table with 84.3%, and Naïve Bayes with 80.1%.
- For stairs down Random Forests outperforms others by showing an accuracy of 95.1% followed by LibSVM with 94.9%, J48 with 91.4%, Decision Table with 91.3%, and Naïve Bayes with 87.3%.
- For sit/stand, Random Forests outperforms others by showing an accuracy of 95% followed by J48 with 90.1%, Naïve Bayes with 83.2%, LibSVM with 81.3%, and Decision Table with 80.1%.
- For collision, Random Forests outperforms others by showing an accuracy of 96.1% followed by J48 with 92.3%, Naïve Bayes with 88.8%, Decision Table 88.2%, and LibSVM with 87.6%.

In the classification of walking activities, only a few basic classifiers, including Naïve Bayes, Random Forests, J48, Decision Table, and LibSVM were used, which on average achieved more than 95% accuracy. Although this accuracy is sufficient in the current scenario for the identification of the mentioned activities, it can be further improved if more advanced classifiers are employed. Examples include extra tree, bagging, AdaBoost, gradient tree boosting as well as deep learning algorithms. We plan to work with these classifiers and compare their results with the current classifiers. Another future research avenue that may further strengthen the research implications of this study is to perform detailed experimentation by involving blind & visually impaired users.

In addition, we can consider multiple disciplines regarding technology advancement and new techniques for future work. The proposed approach can be implemented through the wristband. Wristbands are primarily used for tracking user health and physical activities. These bands also provide a wireless connectivity interface with smartphones and are equipped with a broad range of inertial sensors (accelerometer, gyroscope, and magnetometer) and heartbeat sensors. Smartwatch is another up-to-date general-purpose new technology used by most users to get notifications via smartphone and fitness tracking purposes. Smartwatches are equipped with a broad range of sensors (camera, inertial, global positioning system (GPS), heartbeat, etc.) and processing and wireless connectivity interface with robust storage and power backup. These smartwatches can also provide localization information from GPS and Internet connectivity and user guidance with text-to-speed, vibration, and different sounds or tunes.

Funding This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests The authors explicitly declare “No Competing Interests” are at stake and there is “No Conflict of Interest” with other people or organizations that could inappropriately influence or bias the article’s content.

References

1. Aggarwal JK, Ryoo MS (2011) Human activity analysis: a review. *ACM Comput Surv* 43(3):1–43. <https://doi.org/10.1145/1922649.1922653>
2. Alam F, Pachauri S (2017) Comparative study of J48, naive Bayes and one-R classification technique for credit card fraud detection using WEKA. *Adv Compu Sci Technol* 10(6):1731–1743
3. Andò B, Baglio S, Marletta V, Crispino R, Pistorio A (2019) A measurement strategy to assess the optimal design of an RFID-based navigation aid. *IEEE Trans Instrum Meas* 68(7):2356–2362. <https://doi.org/10.1109/TIM.2018.2879069>
4. Bao L, Intille SS (2004) Activity recognition from user-annotated acceleration data. In: *Proceedings of the Second International Conference on Pervasive Computing, PERVASIVE 2004, Vienna Austria*. Springer, pp 1–17. https://doi.org/10.1007/978-3-540-24646-6_1
5. Blindness and vision impairment: key facts (2021) World Health Organization. <https://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment>. Accessed July 05, 2022
6. Croce D, Gallo P, Garlisi D, Giarré L, Mangione S, Tinnirello I (2014) ARIANNA: A smartphone-based navigation system with human in the loop. In: *22nd Mediterranean Conference of Control and Automation (MED)*, 2014 Palermo, Italy. IEEE, pp 8–13. <https://doi.org/10.1109/MED.2014.6961318>

7. Damaševičius R, Vasiljevas M, Šalkevičius J, Woźniak M (2016) Human activity recognition in AAL environments using random projections. *Computa Mathema Meth Med* 2016:4073584. <https://doi.org/10.1155/2016/4073584>
8. Dargie W Analysis of Time and Frequency Domain Features of Accelerometer Measurements. In: *Proceedings of 18th International Conference on Computer Communications and Networks*, San Francisco, CA, USA., pp 1–6. <https://doi.org/10.1109/ICCCN.2009.5235366>
9. Davis J, Goadrich M (2006) The relationship between Precision-Recall and ROC curves. In: *Proceedings of the 23rd International Conference on Machine Learning (ICML'06)*, Pittsburgh Pennsylvania USA. pp 233–240. <https://doi.org/10.1145/1143844.1143874>
10. Duh PJ, Sung YC, Chiang LYF, Chang YJ, Chen KW (2021) V-eye: a vision-based navigation system for the visually impaired. *IEEE Transac Multimed* 23:1567–1580. <https://doi.org/10.1109/TMM.2020.3001500>
11. Eibe Frank MAH, Ian H (2016) *Data mining: practical machine learning tools and techniques*. 4th edn. Morgan Kaufmann
12. Garcia-Ceja E, Brena R (2013) Long-term activity recognition from accelerometer data. *Procedia Technol* 7: 248–256. <https://doi.org/10.1016/j.proctcy.2013.04.031>
13. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. *ACM SIGKDD Explor Newslett* 11(1):10–18. <https://doi.org/10.1145/1656274.1656278>
14. Hou X, Bergmann J (2021) Pedestrian dead reckoning with wearable sensors: a systematic review. *IEEE Sensors J* 21(1):143–152. <https://doi.org/10.1109/JSEN.2020.3014955>
15. Husin MH, Lim YK (2020) InWalker: smart white cane for the blind. *Disab Rehab: Assist Technol* 15(6): 701–707. <https://doi.org/10.1080/17483107.2019.1615999>
16. Khan M, Ahamed SI, Rahman M, Smith RO (2011) A feature extraction method for realtime human activity recognition on cell phones. In: *Proceedings of 3rd International Symposium on Quality of Life Technology (isQoLT 2011)*, Toronto, Canada, June 5-8. pp. 1–6. https://publications.marquette.edu/mscs_fac/183
17. Khan AM, Siddiqi MH, Lee S-W (2013) Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones. *Sensors* 13(10):13099–13122. <https://doi.org/10.3390/s131013099>
18. Khan AM, Tufail A, Khattak AM, Laine TH (2014) Activity recognition on smartphones via sensor-fusion and KDA-based SVMs. *Int J Distri Sensor Netw* 10:503291. <https://doi.org/10.1155/2014/503291>
19. Khan I, Khusro S, Ullah I (2018) Technology-assisted white cane: evaluation and future directions. *PeerJ* 6: e6058. <https://doi.org/10.7717/peerj.6058>
20. Khan I, Khusro S, Ullah I, Mahfooz S (2019) Towards Smartphone-Based Navigation for Visually Impaired People. In: *Proceedings of the Computational Methods in Systems and Software (CoMeSyso2018)*, Online, Czech Republic. Computational and Statistical Methods in Intelligent Systems. Springer International Publishing, pp 366–373. https://doi.org/10.1007/978-3-030-00211-4_32
21. Khan S, Nazir S, Khan HU (2021) Analysis of navigation assistants for blind and visually impaired people: a systematic review. *IEEE Access* 9:26712–26734. <https://doi.org/10.1109/ACCESS.2021.3052415>
22. Kim SY, Cho K (2013) Usability and design guidelines of smart canes for users with visual impairments. *Int J Des* 7(1):99–110 <http://www.ijdesign.org/index.php/IJDesign/article/view/1209/559>
23. Kunhoth J, Karkar A, Al-Maadeed S, Al-Ali A (2020) Indoor positioning and wayfinding systems: a survey. *Human-centric Comput Inform Sci* 10(1):1–41. <https://doi.org/10.1186/s13673-020-00222-0>
24. Kuriakose B, Shrestha R, Sandnes FE (2022) Tools and Technologies for Blind and Visually Impaired Navigation Support: a review. *IETE Tech Rev* 39(1):3–18. <https://doi.org/10.1080/02564602.2020.1819893>
25. Lee GT, Seo SB, Jeon WS (2021) Indoor Localization by Kalman Filter based Combining of UWB-Positioning and PDR. In: *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, pp 1–6. <https://doi.org/10.1109/CCNC49032.2021.9369588>
26. Li L, Ke Y, Zhang T, Zhao J, Huang Z (2022) A human defecation prediction method based on multi-domain features and improved support vector machine. *Symmetry* 14(9):1763
27. Lin B-S, Lee C-C, Chiang P-Y (2017) Simple smartphone-based guiding system for visually impaired people. *Sensors* 17(6):1371
28. Mahida P, Shahrestani S, Cheung H (2020) Deep learning-based positioning of visually impaired people in indoor environments. *Sensors* 20(21):1–17. <https://doi.org/10.3390/s20216238>
29. Meliones A, Filios C, Llorente J (2022) Reliable ultrasonic obstacle recognition for outdoor blind navigation. *Technol* 10(3):54. <https://doi.org/10.3390/technologies10030054>
30. Mocanu A, Sita V, Avram C, Radu D, Aștilean A (2020) Assistive Navigation Application for Blind People using a White Cane Embedded System. In: *2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, Cluj-Napoca, Romania. IEEE, pp 1–5. <https://doi.org/10.1109/AQTR49680.2020.9129942>

31. Preece SJ, Goulermas JY, Kenney LP, Howard D (2009) A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Trans Biomed Eng* 56(3):871–879. <https://doi.org/10.1109/TBME.2008.2006190>
32. Ravi N, Dandekar N, Mysore P, Littman ML (2005) Activity Recognition from Accelerometer Data. In: Porter B (ed) *IAAI'05: Proceedings of the 17th conference on Innovative applications of artificial intelligence*, Pittsburgh, Pennsylvania. vol 2005. AAAI Press, pp 1541–1546. <https://doi.org/10.5555/1620092.1620107>
33. Ren P, Elyasi F, Manduchi R (2021) Smartphone-based inertial Odometry for blind walkers. *Sensors* 21(12):1–23. <https://doi.org/10.3390/s21124033>
34. Reyes Leiva KM, Jaén-Vargas M, Codina B, Serrano Olmedo JJ (2021) Inertial measurement unit sensors in assistive Technologies for Visually Impaired People, a review. *Sensors* 21(14):1–26. <https://doi.org/10.3390/s21144767>
35. Riehle TH, Anderson SM, Lichter PA, Whalen WE, Giudice NA (2013) Indoor inertial waypoint navigation for the blind. In: 35th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society (EMBC), Osaka, Japan, 03–07 July 2013. IEEE, pp 5187–5190. <https://doi.org/10.1109/EMBC.2013.6610717>
36. Seni G, Elder JF (2010) Ensemble methods in data mining: improving accuracy through combining predictions. *Synt Lect Data Mining Knowle Disc*, vol 2. <https://doi.org/10.2200/S00240ED1V01Y200912DMK002>
37. Shoab M, Scholten H (2013) Havinga PJ Towards physical activity recognition using smartphone sensors. In: *IEEE 10th International Conference on Ubiquitous Intelligence and Computing/Autonomic and Trusted Computing (UIC/ATC)*, 2013, Vietri sul Mare, Italy. IEEE, pp 80–87. <https://doi.org/10.1109/UIC-ATC.2013.43>
38. Su X, Tong H, Ji P (2014) Activity recognition with smartphone sensors. *Tsinghua Sci Technol* 19(3):235–249. <https://doi.org/10.1109/TST.2014.6838194>
39. Vera P, Zenteno D, Salas J (2014) A smartphone-based virtual white cane. *Pattern Anal Applic* 17(3):623–632. <https://doi.org/10.1007/s10044-013-0328-8>
40. Wang HC, Katzschmann RK, Teng S, Araki B, Giarré L, Rus D (2017) Enabling independent navigation for visually impaired people through a wearable vision-based feedback system. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, pp 6533–6540. <https://doi.org/10.1109/ICRA.2017.7989772>
41. Wang Q, Luo H, Wang J, Sun L, Ma Z, Zhang C, Fu M, Zhao F (2022) Recent advances in pedestrian navigation activity recognition: a review. *IEEE Sensors J* 22(8):7499–7518. <https://doi.org/10.1109/JSEN.2022.3153610>
42. Wang Z, Yang Z, Wang Z (2022) An adaptive indoor positioning method using multisource information fusion combing Wi-fi/MM/PDR. *IEEE Sensors J* 22(6):6010–6018. <https://doi.org/10.1109/JSEN.2022.3147309>
43. Wu W, Dasgupta S, Ramirez EE, Peterson C, Norman GJ (2012) Classification accuracies of physical activities using smartphone motion sensors. *J Med Internet Res* 14(5):e130. <https://doi.org/10.2196/jmir.2208>
44. Yasir M, Lestari IN, Setiawan C, Ulfiah EMR, Hamidi EAZ (2021) Design and Implementation of The Blind Navigation Aids Using Ultrasonic Sensor. In: *2021 7th International Conference on Wireless and Telematics (ICWT)*, Bandung, Indonesia, pp 1–6. <https://doi.org/10.1109/ICWT52862.2021.9678455>
45. Zhang Y, Zhang Y, Zhang Z, Bao J, Song Y (2018) Human activity recognition based on time series analysis using U-net. arXiv preprint. <https://doi.org/10.48550/arXiv.1809.08113>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.