

Article

An Adaptive Human Activity-Aided Hand-Held Smartphone-Based Pedestrian Dead Reckoning Positioning System

Bang Wu ^{1,†}, Chengqi Ma ^{2,*}, Stefan Poslad ¹ and David R. Selviah ²

¹ Electronic Engineering and Computer Science Department, Queen Mary University of London, Mile End Road, London E1 4NS, UK; bang.wu@qmul.ac.uk (B.W.); stefan.poslad@qmul.ac.uk (S.P.)

² Electronic and Electrical Engineering Department, University College London (UCL), Torrington Place, London WC1E 7JE, UK; d.selviah@ucl.ac.uk

* Correspondence: chengqi.ma.16@ucl.ac.uk

† Bang Wu and Chengqi Ma contributed equally to this study and share first authorship.

Abstract: Pedestrian dead reckoning (PDR), enabled by smartphones' embedded inertial sensors, is widely applied as a type of indoor positioning system (IPS). However, traditional PDR faces two challenges to improve its accuracy: lack of robustness for different PDR-related human activities and positioning error accumulation over elapsed time. To cope with these issues, we propose a novel adaptive human activity-aided PDR (HAA-PDR) IPS that consists of two main parts, human activity recognition (HAR) and PDR optimization. (1) For HAR, eight different locomotion-related activities are divided into two classes: *steady-heading activities* (ascending/descending stairs, stationary, normal walking, stationary stepping, and lateral walking) and *non-steady-heading activities* (door opening and turning). A hierarchical combination of a support vector machine (SVM) and decision tree (DT) is used to recognize steady-heading activities. An autoencoder-based deep neural network (DNN) and a heading range-based method to recognize door opening and turning, respectively. The overall HAR accuracy is over 98.44%. (2) For optimization methods, a process automatically sets the parameters of the PDR differently for different activities to enhance step counting and step length estimation. Furthermore, a method of trajectory optimization mitigates PDR error accumulation utilizing the non-steady-heading activities. We divided the trajectory into small segments and reconstructed it after targeted optimization of each segment. Our method does not use any a priori knowledge of the building layout, plan, or map. Finally, the mean positioning error of our HAA-PDR in a multilevel building is 1.79 m, which is a significant improvement in accuracy compared with a baseline state-of-the-art PDR system.

Keywords: indoor positioning system; PDR; human activity recognition; machine learning; localization; data fusion; trajectory optimization; parameter optimization; autoencoder neural network; phone pose; movement state

Citation: Wu, B.; Ma, C.; Poslad, S.; Selviah, D.R. A Human Activity-Aided Adaptive Hand-Held Smartphone Based Pedestrian Dead Reckoning Positioning System. *Remote Sens.* **2021**, *13*, 2137. <https://doi.org/10.3390/rs13112137>

Academic Editor: Józef Lisowski

Received: 1 April 2021

Accepted: 26 May 2021

Published: 28 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The indoor positioning system (IPS) has been investigated for several decades for guiding pedestrians around complex buildings on multiple floors such as offices and shopping malls, mining tunnels, and subways, particularly in emergency situations where first responders need to know immediately how to get to those in need when there is limited visibility due to smoke, mist, or having to wear a hazmat or firefighter mask. One of the key features of merit of an IPS is its positioning accuracy, but it must also be robust, giving accurate positions when the users walk in various manners through their environment. Pedestrian IPS can be divided into three categories: (1) radio signal-based IPSs, e.g., using wireless technologies such as WiFi [1–5], ultra-wide band (UWB), [6–8]

and Bluetooth low energy (BLE) [9,10]; (2) optical and acoustic sensor-based IPS, e.g., camera [11–13], visible light [14–16], Lidar [17–19], and sonar ultrasound [20,21] positioning; (3) micro-electromechanical system (MEMS) inertial accelerometers, gyroscopes, and magnetic field sensors [22] used in pedestrian dead reckoning (PDR) [23–25] systems. PDR is a technology that makes use of the length of steps or strides and the heading estimated by an attitude and heading reference system (AHRS) [26,27], fixed on the human body (e.g., foot, waist, and chest) to estimate the next position.

Using PDR to determine position has particular benefits to enable it to be used in the scenarios described. Firstly, it is body mounted and mobile rather than being installed in fixed environmental infrastructure. Secondly, it is independent of the environmental infrastructure, e.g., BLE beacons or Wi-Fi wireless access points (APs) or other location determination transceivers (for example Wi-Fi RTT APs [3]). Thirdly, PDR systems have the advantage that they can work in covered spaces, not just indoors but also underground, and in covered over ground spaces such as under bridges, tunnels, and in built-up spaces, in global navigation satellite systems' signals (e.g., GNSS) denied spaces, or where natural or artificial high-rise structures impede overhead signals. Hence, we make the distinction between position determination in covered versus uncovered spaces and more generally GNSS-denied spaces rather than the current simpler classification of indoor versus outdoor spaces. Fourthly, PDR systems are not affected by wireless interference between a transmitter and receiver. Fifthly, they can be energy-saving if low-cost smartphones with integrated sensors (e.g., accelerometers, gyroscopes, magnetometer, barometers, etc.) are used.

Even though a PDR system has multiple advantages for use in indoor positioning, traditional PDR technology still has issues restricting its use. The first issue is that PDR robustness is strongly affected by human activities with different phone poses [28]. Here, the way in which the phone is carried is called the phone pose (e.g., holding, calling, swinging, and in pocket). Human activity refers to the way the person moves through the environment (e.g., running, walking, ascending, and descending stairs) [29]. Since the same activity has distinctive characteristics for different phone poses [29], the human activity recognition should consider the specific phone pose. Currently, most smartphone-based traditional PDR systems are designed assuming that the user carrying the phone moves forward in a steady walking manner. The PDR robustness is defined as the ability of the PDR method to handle perturbations in steps and headings due to different human activities. Therefore, it is necessary to recognize human activities to improve the robustness of traditional PDR. The second issue is that the positioning accuracy of PDR systems is drastically affected by the accumulated errors of iterative sequential calculations, each of which may have a small error [23]. Therefore, solving the accumulative errors of a traditional PDR system is critical to improving PDR's positioning accuracy. The first category of IPS (e.g., Wi-Fi, BLE, and UWB) can efficiently mitigate the error accumulation of PDR when they are fused with a PDR, but it will increase the hardware cost. In this paper, we postulate that human activities recognized by using smartphone sensor data can be used to improve the accuracy of PDR systems. After human activities are recognized, we can carry out certain optimization methods (e.g., trajectory optimization) to improve the PDR system's positioning accuracy.

To deal with the above issues, we propose a novel adaptive human activity-aided PDR (HAA-PDR) system in this paper. It is adaptive in the sense that the performance of the PDR depends in part upon the human activity, and, hence, adapts to that activity. The framework of the HAA-PDR system is shown in Figure 1a. It includes human activity recognition (presented in Section 3) and optimization (presented in Section 4). Steady-heading activities are used for parameter optimization. Non-steady-heading activities are used for trajectory optimization. In Figure 1b, we give more details about this framework. Our optimization procedure using HAR to PDR is performed in three steps: (1) We parallelly recognize the steady-heading activities and non-steady-heading activities. (2) Parameter optimization is realized by using the identified steady-heading activities to set the

parameters of the step counting and step length estimation methods. (3) After setting the parameters, the trajectory can be constructed. When non-steady-heading activities appear (turning or door opening), we will perform the trajectory optimization accordingly.

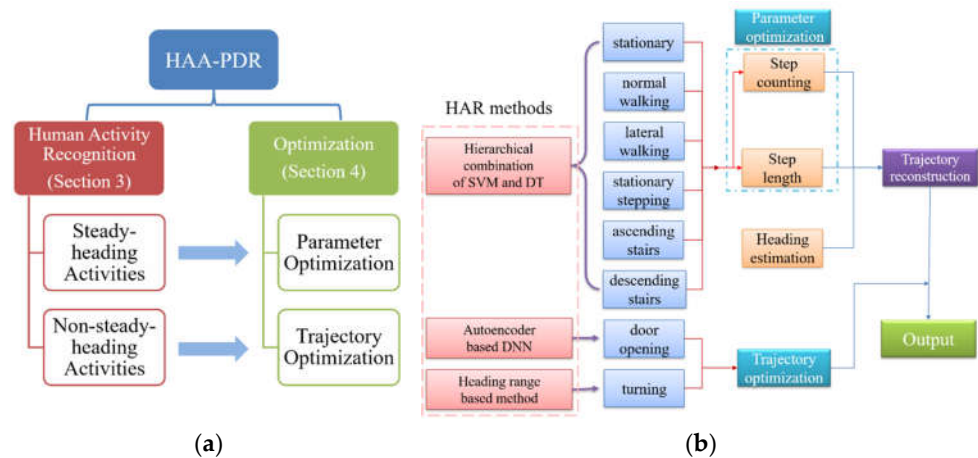


Figure 1. The framework of the HAA-PDR system. (a) is the outline of proposed HAA-PDR system, and (b) is the detailed process flow of this system.

The overview and novel contributions of this work are summarized as follows:

1. We define and recognize two categories of human activities under the phone pose of holding, namely, steady-heading activities (including six sub-activities) and non-steady-heading activities (including two sub-activities) to assist the PDR in improving its accuracy and robustness (see Section 3.1). For steady-heading activities' recognition, a novel hierarchical combination of a support vector machine (SVM) and decision tree (DT) is proposed and demonstrated to improve the recognition performance (see Section 3.2). Two types of non-steady-heading activities are defined and recognized by different methods in parallel and independently, as they have different characteristics. A novel autoencoder-based deep neural network (DNN) is proposed and demonstrated to recognize door opening (see Section 3.3). Meanwhile, a heading range (defined as the difference between the maximum heading and the minimum heading) based method is proposed to recognize (90/180-degree) turning (see Section 3.4).
2. Apart from the proposed methods for activity recognition, two types of optimization are designed to improve the robustness and positioning accuracy of our PDR: parameter optimization and trajectory optimization. Steady-heading activities are used to optimize the parameters in the methods of step counting and step length estimation, respectively, to improve the robustness of PDR (see Section 4.1). Furthermore, a novel method of trajectory optimization is proposed to mitigate PDR error accumulation by utilizing non-steady-heading activities. Unlike conventional sequential PDR processing, we break our trajectory into small segments. Then, the trajectory is reconstructed after targeted optimization of the trace of each activity (see Section 4.2).

The paper is organized as follows: Section 2 introduces the related work. Section 3 introduces the targeted human activities and their recognition methods. Section 4 describes our optimization methods using human activity recognition to aid our PDR to improve robustness and positioning accuracy. Section 5 presents the experimental assessment to validate the performance of our proposed human activity recognition aided PDR system. The last Section 6 gives the conclusions and future work.

2. Related Work

Pedestrian dead reckoning (PDR) technology and human activity recognition (HAR) are two topics that have been widely researched for decades. This section gives a brief overview of these two topics as the background for this research.

2.1. Traditional PDR Technology

Step detection and counting, step/stride length estimation, and heading determination are three crucial processes for PDR positioning systems. For step counting, many accelerometer sensor-based methods have been proposed such as threshold setting [30], peak detection [31], and correlation analysis [32]. Among them, peak detection is most widely used, as it is easy to implement and has a good performance. The main idea of peak detection-based methods is first to detect the acceleration peaks, then to filter out some peaks below a certain amplitude (first threshold), further to filter out some peaks with depending on the difference between them and corresponding valleys below within a certain value (second threshold), and finally to apply a temporal constraint (third threshold) on the detected peaks to reduce overcounting [23,31]. However, these thresholds are different for various movement states or smartphone poses [33]. Adjusting thresholds according to movement states or smartphone poses could improve the performance of step counting.

The second part of the PDR calculation concerns step/stride length estimation. Kim et al. [34] proposed a method using the cubic root of the mean of the acceleration magnitude when walking around a circle to estimate step length in typical human walking behavior. Weinberg [35] thought that the vertical displacement (z-axis) of the human upper body was able to approximate the step length. Therefore, a step length estimation model is built using the quartic root of maximum minus minimum difference of z-axis acceleration. Ladetto [36] observed that the step length and step frequency have a good correlation. He proposed a comprehensive model combining step frequency and variance of the sensor signal. These models could achieve reliable performance in typical walking conditions. However, they still need to be refined to be able to handle special conditions such as ascending/descending stairs and turning.

The inertial sensor orientation or rotation consists of roll, pitch, and yaw in an Euler angle representation, where roll and pitch are also called attitude, elevation, or declination, while yaw is called heading or azimuth angle in the horizontal plane [37]. The process of heading determination is to estimate the orientation. Some methods utilizing gyroscopes or a combination of gyroscopes and accelerometers have been proposed [37,38], but their performance is less robust than the methods using MARG (magnetic, angular rotation rate, and gravity) sensors that consist of gyroscope, accelerometer, and magnetometer sensors. Methods using MARG sensors can estimate the complete orientation relative to the direction of gravity and the earth's magnetic field to reduce the gyroscope's integration error [38]. Kalman filters [39], complementary filters [40], and gradient descent [38] are the major methods used in implementing the orientation estimation based on MARG sensors. After the orientation is calculated, the heading is then found. Though these methods could achieve accurate performance within a short time, the heading drifts away from the heading ground truth over time because of sensor drift. In addition, the heading of the smartphone is sensitive to the movement of the smartphone such as vibration, swinging, and turning. Hence, these factors will drive the trajectory estimated by PDR away from the ground truth. It is necessary to investigate heading correction methods to optimize PDR trajectories.

2.2. Human Activity Recognition Aided PDR Technologies

Accurate human activity recognition with different phone poses is crucial for a robust and accurate PDR system. However, phone pose recognition is only needed if the phone

pose is likely to vary during specific types of human locomotion. The details (human activities, phone poses, and corresponding recognition methods) of the related work are listed in Table 1. In Alessio Martinelli et al.'s work [41], six activities were classified by a relevance vector machine (RVM) that was able to provide probabilistic classification results for each activity. The parameters of the step length estimation algorithm (proposed by Ladetto [36]) for different activities were tuned using a least-square regression method. After RVM outputs the probabilistic classification results, the step length estimation parameters of the two most probable activities were taken to find the weights of the parameters for step length estimation. This operation gives a more robust performance than directly using the parameters of the highest probabilistic activity. Tian et al. [28] recognized three typical phone poses (holding, in-pocket, and swinging) in real time using a finite state machine (FSM). As the directions of the sensor axes were different for different phone poses, this work suggested selecting specific axes according to phone poses to implement existing step counting, step length estimation, and heading determination methods. Mostafa Elhoushi and Georgy [42] proposed a decision tree (DT)-based system to recognize eight activities in [12] phone poses. This work only focused on human activity recognition. No method was proposed to optimize traditional PDR algorithms. Melania Susi et al. [43] proposed a DT-based method to recognize four phone poses. For the specific phone pose of swinging the phone, a step counting method was developed based on the periodic rotation of the arm, as it produced an evident sinusoidal pattern in the gyroscope signal. Beomju Shin et al. [44] recognized three activities (walking, standing, and running) and three phone poses (calling, swinging, and in-pocket) using an artificial neural network (ANN). After recognition of activities and phone poses, the author set different peak thresholds (the first type of thresholds of the peak detection method) to provide more robust step counting. For step-length estimation, the author set a fixed step length for different activities (walking under three different phone poses and running). As for heading estimation, the author assumed there were only four directions that the user moved along. Then, magnetic field data were used to classify these four directions using an ANN. However, the application scenarios of this method are limited, as the moving direction of the user was random. Xu et al. [23] used an ANN to recognize three phone poses. For the step counting method, the author implemented the peak detection method for walking in different phone poses. Then, the authors proposed a step length estimation method based on a neural network and the use of a differential GPS (from smartphone indoors). Considering the heading error during phone pose switching (e.g., a phone maybe handheld but when there is an incoming call, the phone is held close to ear, is no headphones are being used), a zero angular velocity (ZA) algorithm was proposed to correct the heading error caused by such switching. Wang et al. [29] considered the combination of four phone poses and four activities and investigated some machine learning algorithms to recognize these activities. For step counting and step length estimation, the authors thought individual parameters should be set for peak detection and the step length estimation method (proposed by Ladetto [36]) according to movement states. For heading determination, a principal component analysis (PCA)-based method was proposed to remove the heading offset between the pedestrian's direction and smartphone's direction in swinging and in-pocket.

Table 1. Human activities, phone poses, and HAR methods (*walking* means in a relatively straight line).

| Paper | Human Activities | Phone Poses | HAR Methods |
|---------|--|---|--|
| [41] | 6: stationary, walking, walking sideways, ascending and descending stairs, and running | 1: waist-belt | relevance vector machine (RVM) |
| [28] | 3: walking (at slow, moderate, and fast speed) | 3: holding, pocket, swinging | finite state machine (FSM) |
| [42,45] | 6: walking, stationary, ascending and descending stairs, standing/walking on an escalator/moving walkway | 11: holding, backpack, purse, belt holder, chest/arm strapped, hand still by side, dangling, on ear, pocket, wrist, head mount. | Decision Tree (DT) |
| [46] | 7: stationary, walking, running, ascending and descending stairs down elevator, up elevator | 3: pocket, swinging, holding | DT, KNN, SVM, Naive Bayes (NB), linear discriminant analysis (LDA), and least-squares support vector machine (LS-SVM). |
| [43] | 3: stationary, walking, looking for the mobile phone in a bag | 4: swinging, texting, calling, bag | DT |
| [44] | 3: standing, walking, running | 3: swinging, pocket, calling | artificial neural network (ANN) |
| [47] | 7: stationary, standing, normal/fast walking, U-turning, ascending and descending stairs. | 2: swinging, holding | LS-SVM, Bayesian Network using the Gaussian Mixture Model (BN-GMM), DT, LDA, quadratic discriminant analysis (QDA) |
| [23] | 1: walking | 2: holding, swinging | ANN |
| [29] | 4: walking, running, ascending and descending stairs | 4: holding, calling, swinging, and pocket | SVM, DT, MLP, KNN |

For human activity recognition, apart from traditional machine learning methods, many deep learning models have been proposed to overcome the dependence on manual feature extraction and to improve the recognition performance. Zebin [48] et al. proposed a deep convolutional neural network (CNN) model to classify five activities: walking, going upstairs/downstairs, sitting, and sleeping, with an accuracy of 95.4%. Xia [49] et al. proposed a combination of LSTM and CNN architecture for HAR, which was validated using three public datasets (UCI, WISDM, and OPPORTUNITY), achieving accuracies of 95.78%, 95.85%, and 92.63%, respectively. Tufek [50] et al. designed a three-layer LSTM model that was validated using the open dataset (UCI) with an accuracy of 97.4%. Shilong Yu and Long Qin [51] proposed a bidirectional LSTM to recognize six activities based on waist-worn sensor with an accuracy of 93.79%. These examples motivate the application of deep learning models for HAR.

This surveyed work presents examples for combining human activity and phone pose recognition with traditional PDR. To sum up, the studies above can be divided into two categories: the first is to adapt (or optimize) the existing step counting, step length estimation, and heading determination methods according to human activities and phone poses (recognized using machine learning algorithms) to improve the robustness of traditional PDR under these perturbations. The second is to mitigate against step counting, step length estimation, and heading errors caused by transitions between different human locomotion activities or phone poses. However, few studies focus on using human activities to optimize PDR trajectories to reduce the error accumulation. Hence, in our study, we implement and optimize the existing step counting and step length estimation methods in typical human activities (e.g., stationary, normal walking, ascending, and descending). Moreover, we also include recognition of other human activities (door opening and 90/180-degree turning) to optimize the PDR trajectory.

3. Human Activities Recognition

In this section, we proposed our human activities recognition (HAR) method and process in detail with validated recognition accuracy.

3.1. Target Human Activities

The target human activities we chose to study are divided into two groups. The first group is steady-heading activities (defined as activities moving along a relatively straight line without noticeable heading changes). This group is mainly used to tune the parameters of step counting and step length estimation methods. The second group is non-steady-heading activities (defined as activities with certain heading changes), which are used to optimize the PDR trajectories to reduce error accumulation. (1) For the first group of activities, we refer to the activities studied in [41], including standing stationary, walking, walking sideways, ascending and descending stairs, and running. However, we do not consider running, as the user is less likely to run when holding a phone indoors. To avoid ambiguity, we refer to this walking as normal walking and refer to the walking sideways as lateral walking. In addition, we add a new activity called stationary stepping [52], which means the user is taking steps at the same place. This is a form of abnormal walking, which rarely happens in people's normal life. However, it can happen in cases of when someone is cold to warm themselves up in freezing weather or for fitness training. Thus, recognizing this helps to improve the robustness of PDR positioning systems. (2) For the second group of activities, we considered door opening (the user first pulls or pushes the door and then walks through it completely) and turning (only 90-degree and 180-degree turns, as they are the most common turning angles), which are both heading-related activities. The reasons for considering these two activities are two-fold: Firstly, during door opening, the steps generated are disordered, which could aggravate the error accumulation. Secondly, turning usually happens at some specific locations (e.g., the end of corridors, corners of a room), which can be regarded as a landmark [53] feature in the indoor structure to calibrate the PDR. To sum up, eight activities are studied here, including stationary, normal walking, ascending and descending stairs, lateral walking, stationary stepping, door opening, and turning.

Human activity recognition is affected by phone poses, since the same activity may show different characteristics under different phone poses [29]. To understand in detail the effect of perturbations of different human locomotion activities on step length and heading, we just decided in this first major study to fix the phone pose to the most commonly used one (i.e., holding) when walking and navigating. The phone pose, holding, means a user holds a phone horizontally in front of his/her body, as displayed in Figure 2. For navigation purposes, pedestrians usually hold the phone in front of them [29]. Although the phone pose could change when walking, i.e., we move the phone to our pocket or bag because it is unsafe, it is generally far safer to keep holding the phone indoors, as we generally do not walk in mixed transport mode locomotion spaces, i.e., where cars, motorbikes, scooters, cycles, and pedestrians may move in common spaces, e.g., when crossing a road. More phone poses (as shown in Table 1) will be discussed in future work to further improve the PDR performance as phone poses do affect the robustness of PDR. There are a variety of built-in sensors (e.g., accelerometer, gyroscope, magnetometer, barometer light sensor, and sound sensor, etc.) in an off-the-shelf smartphone. Here we only chose accelerometer, gyroscope, magnetometer, and barometer, four of them to recognize target activities. Three sensors, the accelerometer, gyroscope, and magnetometer, have a high sampling frequency (50 Hz) to provide fine-grained source information. The other one, the barometer, is very sensitive to altitude change, although its sample frequency is only about 5 Hz.

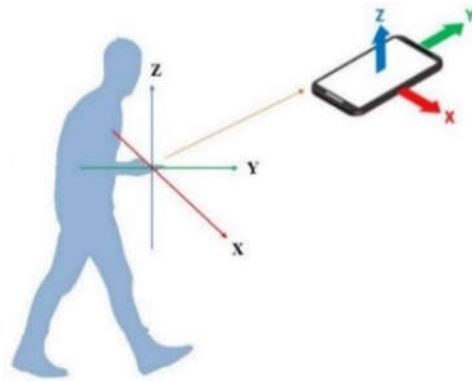


Figure 2. The sketch of using the phone with pose of ‘holding’. The person holds the phone horizontally. The pointing directions of the three orthogonal coordinate axes of the phone frame system are X-axis pointing towards the right, Y-axis pointing towards the front, and Z-axis pointing upwards.

3.2. Steady-Heading Activities Recognition

As we defined them, steady-heading activities include standing stationary, normal walking, ascending or descending stairs, lateral walking, and stationary stepping. To distinguish them, 21 handcrafted features (made by using individual equations instead of using neural networks for automatic feature extraction) are extracted from the four mentioned kinds of sensor data, which are acceleration, gyroscope, magnetometer, and barometer readings. They are the mean of combined magnitude (*mean*), except the barometer; the difference between the maximum and the minimum combined magnitude (*range*); the standard deviation (σ); the skewness of the combined magnitude (*skew*); the kurtosis of the combined magnitude (*kurt*); and the zero-crossing rate (ZCR) of the combined magnitude of acceleration reading [54] (ZCR_{acce}). (Here, we replace the zero in the ‘zero-crossing’ with an average value in ZCR but leave the acronym unchanged, as the average value could be subtracted first.) The mean of the first-order difference of the pressure readings recorded by the barometer is MFD_{pres} ; here, the sampling time interval is fixed and equal. The effectiveness of these features has been investigated in different studies [41,55].

For the classifier, a support vector machine (SVM) has been proven to be able to achieve a high recognition accuracy based on handcrafted features [41,56]. The main idea of SVM is to find a hyperplane to achieve classification. Kernel functions (usually radial basis function kernel) are used to map features from a lower-dimensional space into a higher-dimensional space [57]. In [41], the author used a relevance vector machine (an identical functional form of SVM, but provides a probabilistic classification) to recognize six activities. Hence, we used SVM to recognize our steady-heading activities using two popular methods, Random Forest and KNN, as baselines.

To validate the recognition performance, two young nominally fit male experimenters with differing heights and weights collect for the training and validation dataset. During the data collection stage for training and validation, we decided to only have two experimenters repeatedly performed the activities we were interested in. The data samples from different subjects are roughly equal. All the data samples are mixed up adequately. The reasons we chose two experimenters are that two people can increase the variation of a training dataset, and that it was a heavy workload for only one person to do all the training work. Because a personal smartphone is unlikely to be used by many different people, we did not include many experimenters who have different figures, ages, and habits. In addition, a well-trained model can be applied to more individuals using transfer learning [58], which enables the subsequent personalization learning to be carried out faster either on the phone or in an online connected server. In a real system, more than

two experimenters would be involved, having a diverse range of characteristics, but in this work, two is sufficient to illustrate the concept.

After data collection, all data recordings are equally split into five parts without any overlaps. For each part of the data, a 2 s sliding window with 50% overlap [41] is used for data segmentation. The reason a sliding window was used is that these activities are usually performed continuously sequentially, with one activity merging into the next and not as completely separate discrete activities. A sliding window captures more features from the raw data, including the transition of one activity into the next. As a result, a dataset involving five independent distinct non-overlapping equal-size groups are built. Note that no identical data or instances (defined as the sensor data in each 2 s sliding window) appear in different groups. There are 1526 instances in all, including 255 instances for ascending stairs, 257 instances for descending stairs, 250 instances for stationary, 275 instances for normal walking, 248 instances for lateral walking, and 231 instances for stationary stepping. In each instance, there are 100 data points from the accelerometer, gyroscope, and magnetometer sensors, respectively, and 10 data points from the barometer sensor. The 21 features mentioned above were calculated for each instance. We used a 5-fold cross validation method [59] to avoid overfitting. Each time, we used 4 groups (80% of the whole data) for training and the remaining one group (20% of the whole data) for testing, and we repeated this process 5 times. The recognition performance after 5-fold cross validation is shown in Table 2. The mean accuracy of SVM is 88.71% with a standard deviation (STD) of 0.007, which outperforms Random Forest and KNN.

Table 2. The recognition performance of six activities.

| Classifiers | 5-Fold Cross Validation | Mean | Standard Deviation |
|---------------|--|--------|--------------------|
| SVM | 88.79%, 88.79%, 87.93%, 90.09%, 87.93% | 88.71% | 0.007 |
| Random Forest | 78.02%, 85.34%, 80.60%, 78.88%, 78.45% | 80.26% | 0.026 |
| KNN | 82.76%, 82.33%, 82.33%, 81.90%, 81.90% | 82.24% | 0.003 |

The corresponding confusion matrix is displayed in Table 3. The first three activities (stationary, ascending, and descending stairs) have very good accuracy; only O2 has an 8.33% chance of wrongly being recognized as O4 normal walking. However, the prediction accuracy of the normal walking, lateral walking, and stationary stepping activities are rather low. From the result, we can find that O4 normal walking, O5 lateral walking, and O6 stationary stepping are easily confused from each other by the SVM, because they have similar features. Hence, we need to figure out a better way to distinguish them.

Table 3. The confusion matrix of six activities: stationary (O1), ascending (O2), descending (O3), normal walking (O4), lateral walking (O5), and stationary stepping (O6).

| | Predicted Activities | | | | | |
|----|----------------------|--------|------|--------|--------|--------|
| | O1 | O2 | O3 | O4 | O5 | O6 |
| O1 | 100% | 0% | 0% | 0% | 0% | 0% |
| O2 | 0% | 91.67% | 0% | 8.33% | 0% | 0% |
| O3 | 0% | 0% | 100% | 0% | 0% | 0% |
| O4 | 4.35% | 0% | 0% | 82.61% | 13.04% | 0% |
| O5 | 0% | 0% | 0% | 26.19% | 71.43% | 2.38% |
| O6 | 0% | 0% | 0% | 15.38% | 5.13% | 79.49% |

Since the O4, O5, and O6 are quite similar to each other, we tried to merge them into one type of activity, called combined walking. We retrain the SVM model to recognize four new activities: stationary, ascending, descending, and combined walking.

SVM shows a better performance after we merged O4, O5, and O6. The number of instances for stationary, ascending, and descending remains the same. We shuffled the

instances of normal walking, lateral walking, and stationary stepping and randomly select 1/3 of these instances in a stratified sampling way so that the instances for combined walking are 251. The reason for this operation is to reduce the sample imbalance of the four new activities. Then the recognition performance after 5-fold cross validation is listed in Table 4. The mean accuracy of SVM is 98.41% with a standard deviation (STD) of 0.002, which outperforms Random Forest and KNN. In Table 5, we can find the confusion matrix showing 100% accuracy, except for activity O1 and activity O2, which show 97.96% and 95.65%, respectively. As a result, it is noticeable that the recognition accuracy significantly improves (by 9.7% for the mean accuracy). Thus, we firstly used SVM to recognize stationary, ascending, descending, and combined walking. The next step is to further discriminate normal walking, lateral walking, and stationary stepping.

Table 4. The recognition performance of merged four activities.

| Classifiers | 5-Fold Cross Validation | Mean | Standard Deviation |
|---------------|--|--------|--------------------|
| SVM | 98.30%, 98.30%, 98.30%, 98.42%, 98.72% | 98.41% | 0.002 |
| Random Forest | 94.89%, 97.45%, 97.87%, 95.32%, 97.87% | 96.68% | 0.013 |
| KNN | 97.02%, 96.60%, 95.32%, 97.02%, 97.45% | 96.68% | 0.007 |

Table 5. The confusion matrix of four activities: stationary (O1), ascending (O2), descending (O3), and combined walking (Ns).

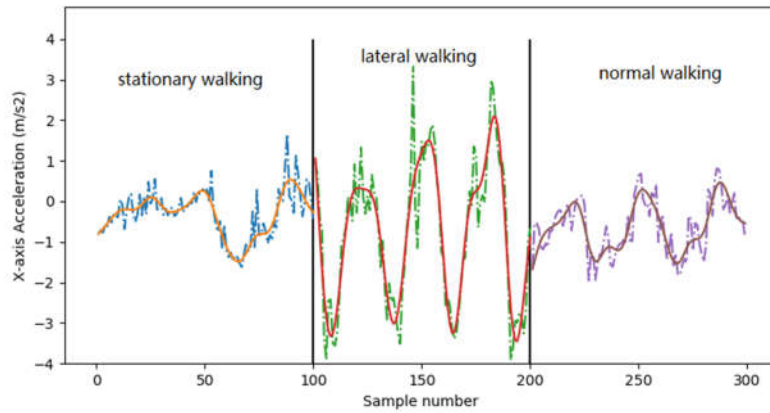
| | Predicted Activities | | | |
|----|----------------------|--------|------|-------|
| | O1 | O2 | O3 | Ns |
| O1 | 97.96% | 0% | 0% | 2.04% |
| O2 | 0% | 95.65% | 0% | 4.35% |
| O3 | 0% | 0% | 100% | 0% |
| Ns | 0% | 0% | 0% | 100% |

Then, we applied a new classifier decision tree (DT) to further recognize the combined walking. We studied the X-axis and the magnitude acceleration of stationary stepping, lateral walking, and normal walking in Figure 3. We employed a low-pass filter (order = 6, cut-off frequency = 2 Hz) to remove noisy glitches. The dash-dotted lines (in blue, green, and purple) with many glitches are the raw measurements, and the much smoother solid lines (in yellow, red, and tan/brown) are the filtered measurements. In Figure 3a, the left part split by the vertical back line belongs to stationary stepping, the middle part is lateral walking, and the right part is normal walking. We can see that the fluctuation in the X-axis acceleration for lateral walking is larger and has a clear regular pattern than that of stationary stepping and normal walking, as the direction of movement is lateral towards the left/right hand side. In Figure 3b, we display the magnitude of acceleration for different types of walking. It is noticeable that the change in this magnitude of acceleration for stationary stepping is clearly less than that of lateral walking and normal walking. Therefore, we can make use of X-axis acceleration to distinguish lateral walking from stationary stepping and normal walking. Meanwhile, we can use this magnitude of acceleration to distinguish stationary stepping from lateral walking and normal walking. As a result, three types of walking can be classified by using two features, $\sigma_{acce,x}$ and $mean_{acce}$, extracted from the magnitude of acceleration and X-axis acceleration, respectively.

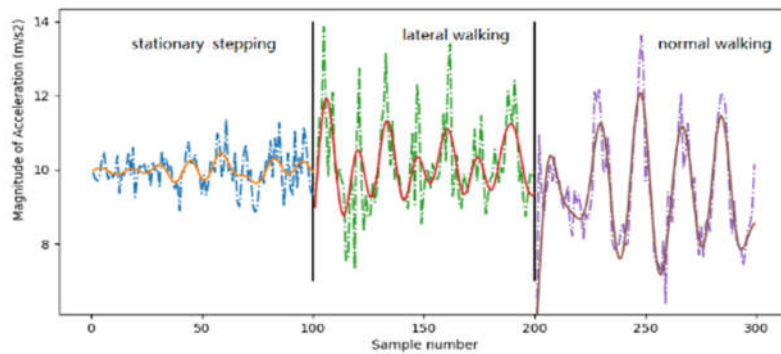
$$mean_{acce} = \frac{1}{N} \sum_{i=1}^N \sqrt{x_i^2 + y_i^2 + z_i^2} \quad (1)$$

$$\sigma_{acce,x} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(x_i - \frac{1}{N} \sum_{i=1}^N x_i \right)^2} \quad (2)$$

where x_i, y_i, z_i are i -th acceleration samples of the X-axis, Y-axis, and Z-axis, respectively. N is the number of acceleration samples within a 2 s window.



(a)



(b)

Figure 3. (a) The change of X-axis acceleration and (b) the magnitude of combined acceleration for stationary stepping, lateral walking, and normal walking (sampling frequency: 50 Hz).

We chose the decision rule-based decision tree (DT) as our classifier to recognize three types of walking and compared this to two popular baselines, Random Forest and KNN. The flowchart of this method is shown in Figure 4. Firstly, two features, σ_{acce_x} and $mean_{acce}$, are extracted from the filtered accelerometer data. Secondly, three rules are designed to discriminate stationary stepping, normal walking, and lateral walking.

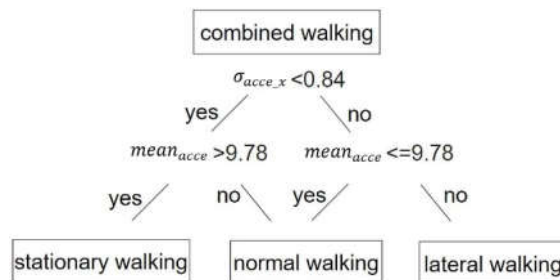


Figure 4. The flow chart of DT-based classification for normal walking, stationary stepping, and lateral walking.

To validate the recognition performance, we formed a dataset with 754 instances: 231 instances for stationary stepping, 248 instances for lateral walking (including leftward and rightward), and 275 instances for normal walking. The recognition performance after 5-

fold cross validation is shown in Table 6. In conclusion, the mean accuracy of DT on recognizing normal walking, lateral walking, and stationary stepping is 99.44%, which outperforms Random Forest and KNN.

Table 6. The performance of DT classifier.

| Classifiers | 5-Fold Cross Validation | Mean | Standard Deviation |
|---------------|--|--------|--------------------|
| DT | 99.07%, 99.07%, 1.0, 1.0, 99.07% | 99.44% | 0.005 |
| Random Forest | 99.07%, 99.07%, 1.0, 99.07%, 99.07% | 99.25% | 0.004 |
| KNN | 97.33%, 99.07%, 99.07%, 96.32%, 97.19% | 97.79% | 0.011 |

To sum up, the approach we used to recognize six steady-heading activities is the hierarchical combination of SVM and DT as shown in Figure 5. First, SVM was used to recognize stationary, ascending, descending, and combined walking. Then, DT was used to further discriminate normal walking, lateral walking, and stationary stepping. The experimental performance shows that this combination has a better performance by over 9% than using SVM alone. The significance of this approach is that when the recognition accuracy of the six activities was insufficient, we performed a deep investigation of the performance of each activity and used a hierarchical combination of two traditional machine learning algorithms to improve the recognition accuracy instead of significantly increasing the complexity of the recognition model as some popular deep learning algorithms do, e.g., CNN, VGG, ResNet, Inception, Xception, etc. Moreover, our approach achieves very high recognition accuracy using only a few samples, so it is very efficient.

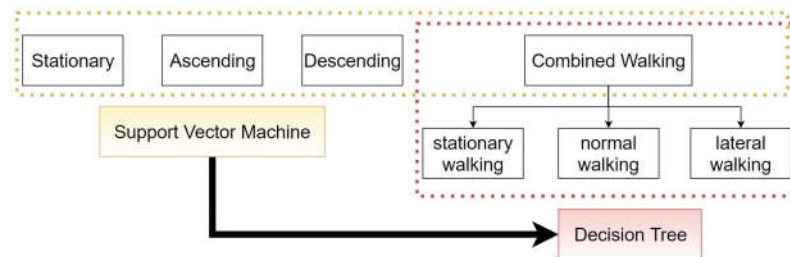


Figure 5. The hierarchical combination of SVM and DT to recognize heading steady activities.

3.3. Non-Steady-Heading Activity Recognition: Door Opening

For non-steady-heading activities' (door opening and turning) recognition, we used different methods for recognition, since door opening is a complicated process including pulling/pushing a door, heading change, and disordered steps. In contrast, turning is more related to heading changes but not to step changes. The activity of door opening typically includes two stages: pushing or pulling a door and then stepping through it. Note that the action of closing a door is not considered for the doors in this study because most doors in modern buildings close themselves automatically.

There are two noteworthy features for this activity. The first feature is the variation in the combined magnitude of acceleration. We made a comparison between door opening and normal walking in Figure 6. The blue solid line is the combined magnitude of acceleration during normal walking. The orange dashed line is the combined magnitude of acceleration during door opening. The fluctuation range of the blue solid line is about 6–14 m/s^2 . The fluctuation range of the orange dashed line is about 8–12 m/s^2 . This difference is the first feature. The second feature is a noticeable heading change during door opening, as shown in Figure 7b, since we usually walk around a door after we pull or push it into a half-open position, as shown in Figure 7a. Although there are a variety of ways to open a door, we only investigate this most common way in our research. Note that Figure 7a shows that the heading is changing during this activity, which is considered as a non-steady-heading activity (activities with certain heading changes). However, we

consider that the heading directions at the beginning and ending time points should be the same.

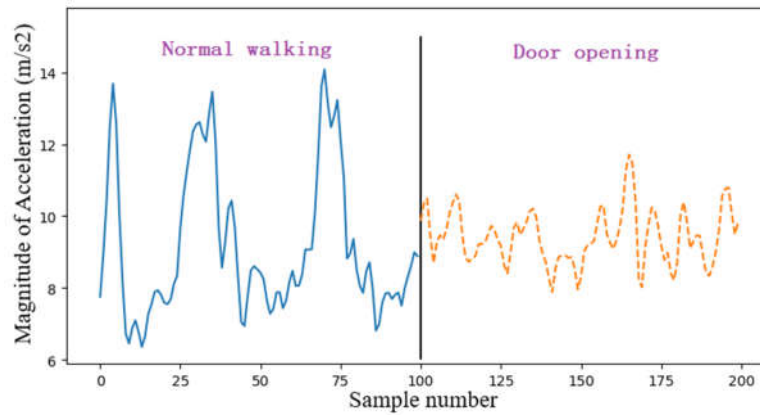


Figure 6. Fluctuation of the magnitude of accelerometer for door opening and normal walking.

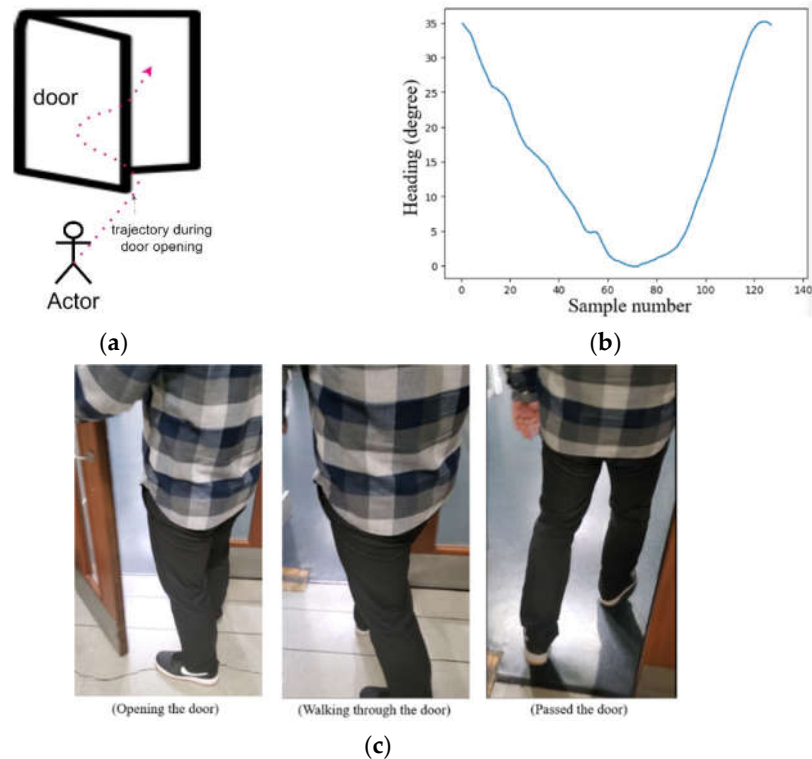


Figure 7. One example of the change of heading when the user performs door opening. (a) demonstrates the process of opening a door and going through it. The red arrow indicates the change of heading in this process. (b) The corresponding change of heading numerically. (c) Pictures of the door opening activity showing a heading change.

To further analyze the characteristics of heading change, an experiment of door opening is conducted. There are 16 doors including wooden fire doors and steel fire doors (all of them can be closed automatically) for the two experimenters to perform opening and going through the door. Half of the experiments were pulling the door and then crossing the door, as shown in Figure 7c, and the other half were opening and then crossing the door. This experiment was repeated 10 times. Based on our collected samples, the time cost for door opening each time was more than 2.0 s. Thus, we empirically selected a 100-

element ($2.0 \text{ s} \times 50 \text{ Hz} = 100$) window to segment the heading data, as shown in Figure 8a. The blue solid line is the heading, and the rectangles are the segmentation windows. First, for each window, after the start and the end are selected, the first heading (or start heading) should equal the last heading (or end heading), because after the user passes the door, he should return to the direction he was in when he started to open the door. However, in practice, the first heading does not always equal the last heading. To mitigate this deviation, we set a threshold for the difference between the first heading and last heading. However, the choice of this difference threshold causes a trade-off in the heading characteristic of door opening. If this threshold is very small (e.g., close to 0 degrees), only a few windows can represent the door opening, which increases the difficulty in collecting enough training data. If this threshold is very large (e.g., 30 degrees), too many windows represent the door opening, which increases the difficulty in accurately recognizing them. Therefore, we set this difference threshold to be 8 degrees empirically in our work.

In Figure 8a, we show three example rectangular windows to segment the heading sequence (the blue solid line). In addition, we set another threshold for the change in heading range to only focus on noticeable heading fluctuations, which distinguish door opening from other activities with slight heading fluctuation. This threshold was set as 20 degrees after statistical analysis of the heading range for all our examples. Thus, for a heading sequence, we segmented it from its beginning by using a 100-element window and moved this 1 sample at a time. For each segmentation, if the difference between the first heading and last heading is no more than 8 degrees and heading range of this segmentation is no more than 20 degrees, this segmentation represents door opening, otherwise it does not. After finishing segmentation, we could, for example, reverse the data so that entering a door from the left appears to mimic entering from the right (as shown in Figure 8b, and we can also do this vice versa too) to augment the dataset, assuming left opening is a reflection of right opening. We confirm that this assumption is valid by comparing the reversed data with experimental heading changes for left opening doors.

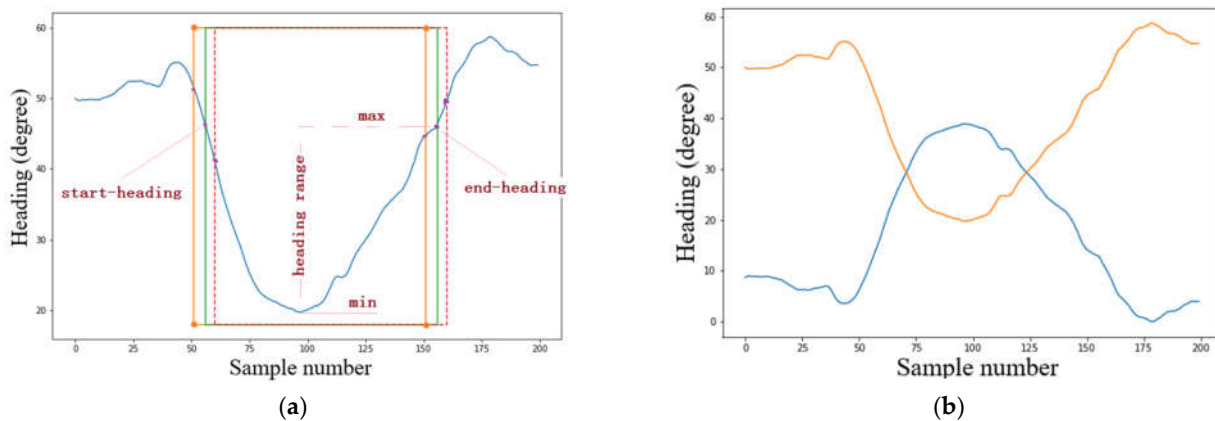


Figure 8. (a) Showing the segmentation of the heading sequence using a fixed size window. The orange dotted rectangle is the heading window in which the first heading is about 8 degrees more than the last heading. The red dashed rectangle is the heading window in which the last heading is about 8 degrees more than the first heading. The green valid rectangle is the heading window in which the first heading equals to the last heading. (b) The reverse of the heading sequence from left to right to create additional training data.

The acceleration changes at the same time as the heading changes so the magnitude of acceleration is segmented in the same way as the heading. There are 100 heading elements and 100 magnitudes of acceleration elements in each window. In our dataset, there are about 22,000 segments for door opening and 22,000 segments for non-door opening. It is a binary classification problem. We propose an autoencoder-based deep neural network to recognize door opening. The reason we chose an autoencoder is that a deep neural network can automatically learn features from a dataset in an unsupervised way [60]. The

autoencoder-based feature extraction is more efficient and powerful than hand-crafted features. The process of learning internal representations [61,62] from the input data is called coding (or encoder), and the process of using such an internal representation to reconstruct the original input data is called decoding (or decoder).

The structure of our model is depicted in Figure 9, where two phases are involved. In the first phase, the autoencoder (including encoder and decoder) was trained (using back propagation) to obtain the internal representation. In the encoder, the input is the combination of heading and magnitude of acceleration. Two 2D convolutional layers (are used to extract features as they have a good performance for feature extraction) and one dropout layer (is used to alleviate overfitting) were added to obtain the internal representation of the input. Even though more convolutional layers could be employed, considering calculation load and efficiency, we only used two convolutional layers. The pooling layers were not used. This keeps the features of heading and acceleration independent for the same type of activities. In the decoder, three transposed 2D convolutional layers and one dropout layer were used to reconstruct the original input data. The optimizer we used for the training is Adam with a learning rate of 0.001, as Adam [63] is an adaptive moment estimation that can compute the adaptive learning rate for parameters so that a faster convergence and a higher learning efficiency can be achieved than others. In the second phase, after the autoencoder was trained, the encoder was combined with the classifier to be further trained for classification. Two dense layers (used to gather all extracted feature) and one SoftMax layer were added in this classifier. The number of neurons in Dense 2 is 30, which means we empirically chose 30 features for classification.

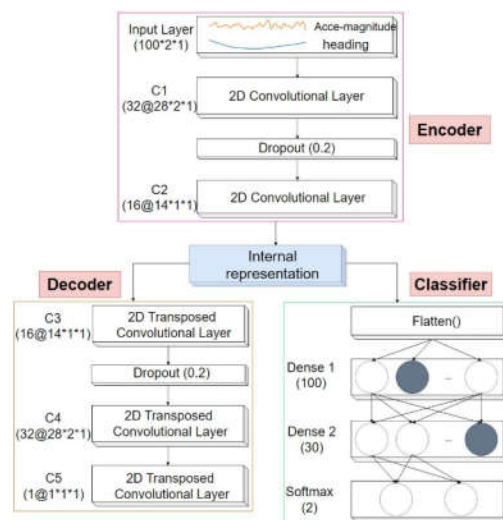


Figure 9. The structure of autoencoder-based deep neural network.

Our dataset is split into a training dataset (70%), a validation dataset (20%), and a testing dataset (10%). The performance metrics of our classifier for door opening and non-door opening are listed in Table 7. The recognition accuracy, recall, precision, and F1_score [64] of door opening are 99.86%, 99.84%, 99.86%, and 99.85%, respectively. In addition, the normalized true positive, false positive, true negative, and false negative are 99.95%, 0.05%, 99.77%, and 0.23%, respectively.

Table 7. Recognition performance of door opening.

| Metrics | Accuracy | Recall | Precision | F1_Score |
|---------|----------|--------|-----------|----------|
| Value | 99.86% | 99.84% | 99.86% | 99.85% |

Figure 10 shows the training and validation's loss and accuracy of the classifier over training epochs in (a) and (b), respectively. The blue solid lines represent the training set and the orange dashed lines represent the validation set. The number of training epochs is 600.

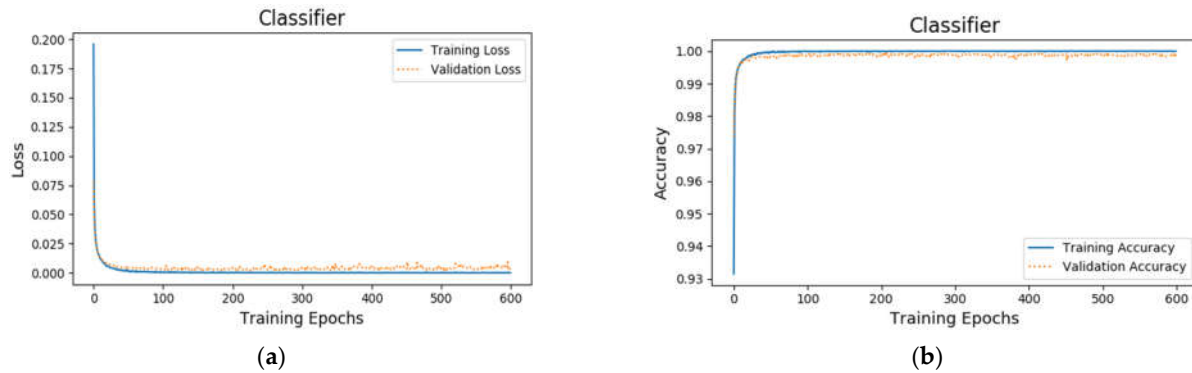


Figure 10. (a) The training and validation's loss, and (b) the accuracy of the classifier over training epochs.

3.4. Non-Steady-Heading Activities Recognition: Turning

Turning usually takes place at some specific locations such as the corners of a room, ends of a corridor, entrances, and exits to and from a staircase. They can be viewed as landmark [53] features in the indoor structure to calibrate the PDR positioning. In our work, we only distinguish 90-degree and 180-degree turning. Ninety-degree turns tend to occur in human-designed indoor environments (e.g., when walking around a corner in a corridor), whereas 180-degree turns can occur when people turn around, for example, when someone finds that he is walking in a wrong direction or to a dead-end corridor.

To detect turning activities, a gyroscope-based threshold setting algorithm [65] is a typical method used to do this, since the gyroscope can accurately measure the angular turning rate in a short time. This method is implemented by setting a threshold for the magnitude of the gyroscope reading. If the magnitude of the gyroscope reading is more than this threshold, turning is detected. However, since the angular turning rate determines the speed of turning (not the amount of change in heading), it is intrinsically hard to discriminate between small-angle turning and large-angle turning. Thus, we propose a novel heading-range method to improve the turning recognition accuracy. The idea is that if the heading change in a fixed period is more than a threshold, a turning is detected. To analyze the characteristics of heading change during turning, we collected data of 138 90-degree and 180-degree turns. Three examples are shown in Figure 11.

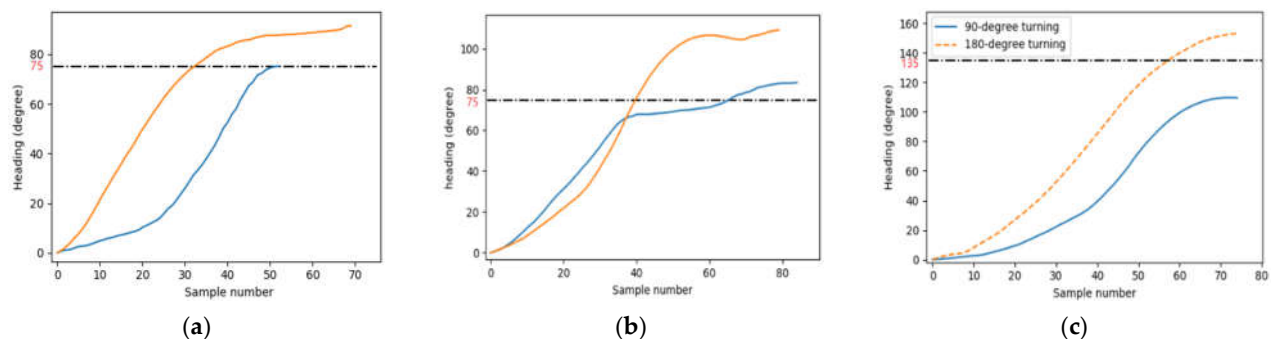


Figure 11. Examples of heading change while performing 90-degree and 180-degree turning. (a) Ninety-degree turning examples in C_1 , (b) 90-degree turning examples in C_2 , and (c) 90-degree and 180-degree turning in a 75-element window.

Based on our experiment, the average time taken for 90-degree turning is about 1.5 s. There are 75 elements, as the frequency is 50 Hz, so we used a window with size of 75 heading elements to detect turning. To determine the threshold for turning, we need to calculate the minimum heading range of those 138 turnings. We divide those turnings into two categories based on the time cost for turning, C_1 (turning time less than 1.5 s) and C_2 (turning time more than 1.5 s). For each turning in C_1 , we directly calculate its heading range from all heading elements. For each turning in C_2 , as the number of heading elements is more than 75, we used a 75-element window (moving from, then through, the first heading element one at a time) to calculate the heading range of those chosen heading elements. As there are multiple windows applied for each turning in C_2 , we calculate the heading range of each window and take the maximum as the heading range of current turning. In this way, we can calculate heading ranges for all turnings in C_2 . Then, we put the heading ranges of those 138 turns together and find that the minimum heading range is 75 degrees, which is shown by the black dash-dotted horizontal line in Figure 11a,b. Thus, we set 75 degrees as the threshold for our method.

After turning has been detected by our heading range-based method, we need to further discriminate 90-degree and 180-degree turns. We take 135 degrees (middle value, the black dash-dotted horizontal line as shown in Figure 11c) between 90 and 180 degrees as the second threshold. The process is: if the heading range of turning is more than 135 degrees, we judge it to be a 180-degree turning, or if not, we judge it to be a 90-degree turning. The pseudo code that implements our approach is shown in Figure 12.

```

Input heading_sequence: [h1, h2, ..., hi, ..., hN]
       time_sequence (the time each heading element sampled): [t1, t2, ..., ti, ..., tN]
window_size = 75
threshold = 75 degrees
rule = 135 degrees
turning_heading = [] #save heading elements during turning
turning_time = [] #save time of heading elements during turning
for i in 1, 2, 3... (N - window_size):
    [Turning detection]
    heading_window = heading_sequence[i : i + window_size]
    if max(heading_window) - min(heading_window) >= threshold :
        output "turning detected"
        turning_record.add(i)
    else
        if length(turning_record) != 0
            start_index = turning_record[0]
            end_index = turning_record[-1] + window_size + 1
            turning_heading = heading_sequence[start_index: end_index]
            turning_time = time_sequence[start_index: end_index]
    [Turning detected, further discriminate 90 and 180 degree turning]
    if max(turning_heading) - min(turning_heading) < rule :
        output "it is a 90 degree turning"
    else
        output "it is a 180 degree turning"
turning_record = []

```

Figure 12. The pseudo code of turning detection discriminates 90-degree from 180-degree turning. Note: the heading is computed using a complementary filter method, as it is stable, efficient, and easy to implement [40].

4. Adaptive Human Activity-Aided Pedestrian Dead Reckoning

Based on the human activity recognition (HAR) results, our adaptive human activity-aided pedestrian dead reckoning (HAA-PDR) system is designed. As shown in Figure 1, the HAR results are used in two parts of the system, parameter optimization and trajectory optimization.

4.1. Parameter Optimization Based on Steady-Heading Activities

Parameter optimization is a process to set the best values for the parameters of the PDR methods for each different activity, to improve the robustness of the PDR system. The PDR methods studied here are step counting and step length estimation. The idea of this parameter optimization refers to the work of Wang et al. [29] for step counting and Shin et al. [44] for step length estimation.

4.1.1. Step Counting

We used the peak detection method for step counting, as this method is widely used in smartphone-based PDR positioning [31]. Three thresholds were needed to implement this method. Thus, we considered the threshold setting for each activity.

- (1) *Three thresholds*: In the peak detection method, there are three types of thresholds, as shown in Figure 13. The red points are peaks and the green points are valleys. The orange dashed curve is the magnitude of accelerometer readings. The first threshold is the peak threshold (shown by the horizontal line in Figure 13). A peak is detected if its value is above this threshold. The second threshold is the difference between the combined magnitude of acceleration peak and its adjacent valley, $diff_{threshold}$. When that difference is more than the threshold, a step is counted [66]. The third threshold is the time interval between two sequential steps, $\Delta t_{threshold} = 0.2$ s (shown by red points) according to the studies [67,68], when a pedestrians' step frequency is less than 5 Hz (0.2 s per step). A step is counted only if the time interval is more than this threshold.

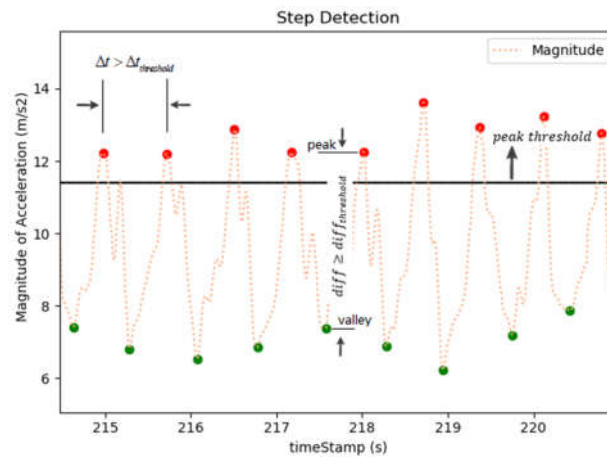


Figure 13. Three thresholds of the peak detection method.

- (2) *Threshold setting for each activity*. The activities we focus on are normal walking, lateral walking, ascending stairs, and descending stairs. Since the amount of the change of accelerometer reading is different in each of these four activities, different thresholds are set for each different activity. To find the performance of different threshold sets of values, we collected step data from two young male nominally fit experimenters of differing heights and weights performing normal walking, lateral walking, descending stairs, and ascending stairs. In this experiment, two experimenters held their smartphones (two different phones, Nexus 5 and Google Pixel 2) to collect the steps for the targeted activities. The raw data were processed using a low-pass filter (order = 6, cut-off frequency = 5 Hz) before step counting. The parameters of this low-pass filter were set empirically based on our experimental data. The performance details of step detection are shown in Table 8, where peak, time, and diff are abbreviations for the peak threshold, time interval threshold, and difference threshold, respectively. After setting different thresholds for each activity, the mean step accuracy

(the calculation equation refers to the equation (34) in the work of Poulouse et al. [24]) achieves 99.25% (from Table 9). We can conclude that setting individual thresholds for different activity patterns gives a very good counting accuracy. Thus, once we recognized these activities, we set corresponding different thresholds for each of the different activities.

Table 8. The performance details of step detection for different activities.

| Activities | Thresholds (Peak; Time; Diff) | Real steps | Predicted steps | Accuracy (%) |
|-------------------|----------------------------------|---------------|--------------------|-----------------|
| Normal walking | 11; 0.2; 1.5 | 425 | 427 | 99.53 |
| Lateral walking | 10.58; 0.2; 2 | 400 | 406 | 98.5 |
| Ascending stairs | 11.4; 0.2; 1.5 | 198 | 199 | 99.49 |
| Descending stairs | 11.4; 0.2; 1.5 | 198 | 197 | 99.49 |

Table 9. Details of positioning accuracy assessment for different process strategies.

| Strategies | Mean (m) | Standard Deviation (m) | Error at Final Position (m) | CDF (m) | |
|--|----------|---------------------------|--------------------------------|---------|-------|
| | | | | 95% | 85% |
| Baseline | 37.73 | 29.46 | 63.72 | 76.05 | 72.21 |
| Door-Opening-Based Optimization | 36.82 | 28.95 | 58.39 | 75.25 | 71.39 |
| Turning-Based Optimization | 1.92 | 1.49 | 5.27 | 5.27 | 3.00 |
| HAA-PDR | 1.79 | 1.13 | 2.68 | 3.72 | 2.98 |

4.1.2. Step Length Estimation

The step length depends on which human activities are being performed. Here, we considered step length estimation for the activities of stationary, walking (including normal walking, stationary stepping and lateral walking), turning, ascending stairs, and descending stairs. The basic step length estimation method we used here is the Weinberg method [35] defined in equation (3), where z_{max} , z_{min} are the maximum and minimum of Z-axis accelerometer readings (during a step), and λ is a parameter to be set.

$$S = \lambda * 0.34 * \sqrt[4]{z_{max} - z_{min}} \quad (3)$$

According to the activities we discussed in this paper, we set parameters individually for the step length of each activity. The parameters for step length estimation in different activities are listed as follows:

- (1) Stationary: When a user is stationary, the step length is 0, i.e., $\lambda = 0$.
- (2) Combined Walking: It includes normal walking, stationary stepping, and lateral walking. Normal walking and lateral walking have the same step length parameter, $\lambda = 1$. For stationary stepping, there is no displacement even if it has similar accelerometer characteristics as normal walking; thus, $\lambda = 0$.
- (3) Ascending or descending stairs: When a user ascends stairs or descends stairs, the step length usually equals the width of the stair. The stair width in the experimental environment is 30 cm. Thus, we set the step length to a constant value of 0.3 m, $S = 0.3$.

4.2. Trajectory Optimization Based on Non-Steady-Heading Activities

Trajectory optimization is a process for improving the trajectories estimated by PDR by combining them with information about which human activity is being performed to mitigate the error accumulation of PDR. The activities studied here are door opening and turning.

4.2.1. Door Opening

The effect of door opening on PDR positioning is very complicated. The steps generated during door opening are disordered, so it is hard to determine the correct trajectory during door opening.

For example, in Figure 14a, a person walks from position A to position B and starts to pull the door at B, then passes through the doorway to position C to finish door opening and keeps walking to destination D. The user's displacement during door opening is 0.5 m (the straight-line distance from B to C). The coordinates calculated by PDR are shown in Figure 14b. The black solid line (A→B→C→D) is the trajectory estimated by PDR. The yellow starred line (B→C) is the trajectory during door opening. Obviously, the yellow starred line does not display the real displacement during door opening, as the steps generated are disordered. We need to optimize this trajectory to improve positioning performance. This involves extending the line segment (O→B) to a point C' with a condition that the distance (B→C') is 0.5 m (the user's displacement during door opening). Here, point O is the closest location to point B before door opening. Hence, the line segment (B→C') represents the corrected trajectory during door opening. Position D' is the correction of position D. The red dashed line (B→C'→D') is the optimized trajectory (B→C→D). The reason we optimized the estimated trajectory in this way is that a user usually keeps walking along the same direction (from O to B) if there is no door opening. The equivalent displacement during door opening is 0.5 m. Obviously, if a user suddenly turns left or right after door opening, our method still works, as the user needs space to turn, so the user usually moves 0.5 m in the process.

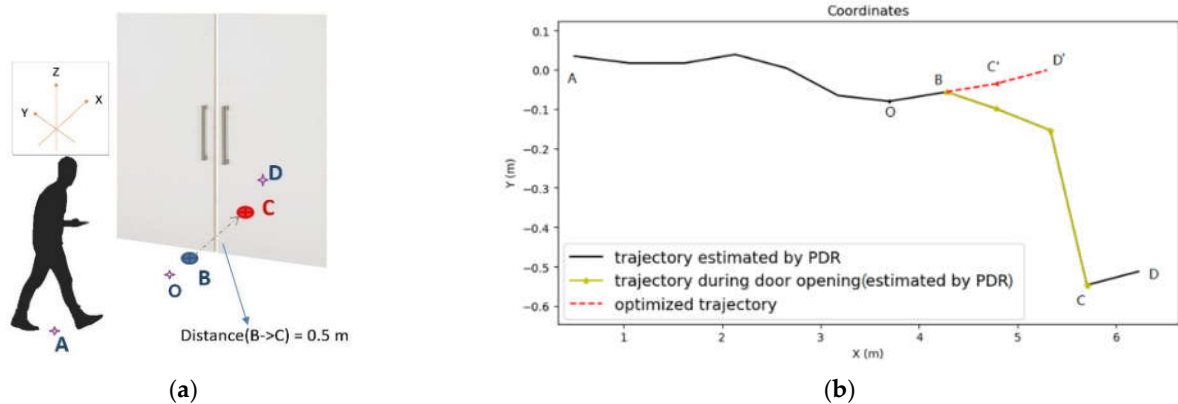


Figure 14. (a) The illustration of door opening trajectory. (b) An example of trajectory optimization during door opening.

4.2.2. Turning

We proposed making use of the characteristics of non-steady-heading activities to mitigate this problem. Our idea was to adjust the angle between the trajectories of the subsequent activity and the previous activity based on the type of current non-steady-heading activity.

As shown in Figure 15, a person moves from the start point A to the end point C along the red dotted line. This person performs normal walking from A to B, then makes a 90-degree left turning at the turning point B and performs normal walking from B to C. We split this path into sub-trajectories based on turning: the sub-trajectories \widehat{AB} and \widehat{BC} . In practice, due to the heading error, the sub-trajectories we calculated are \widehat{AB} and $\widehat{BC'}$, which are shown as fine dark blue solid lines. Obviously, the sub-trajectory $\widehat{BC'}$ should be corrected to the sub-trajectory \widehat{BC} . In this case, the turning angle at B is 90 degrees. This means the heading difference between the sub-trajectories AB and BC should be 90 degrees. For the curved sub-trajectory like \widehat{AB} , we used the heading of the line segment to

connect the first point A and the end point B as the heading of \widetilde{AB} , noted as $heading_{AB}$. Similarly, the headings of $\widetilde{BC'}$ and \widetilde{BC} are noted as $heading_{BC'}$ and $heading_{BC}$.

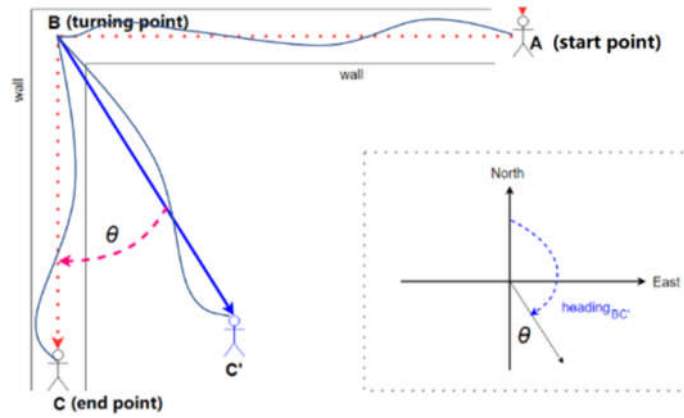


Figure 15. A sketch of heading optimization based on non-steady-heading activities (such as turning).

Our solution to correct $\widetilde{BC'}$ into \widetilde{BC} is to rotate $\widetilde{BC'}$ clockwise through an angle, θ . Hence, our proposed heading optimization has two steps. The first step is to calculate this rotation angle, θ , at the current turning point B using Equation (4) or (5) after recognizing that the human activity of 90-degree left turn is taking place and where and when it is taking place. The second step is to rotate the estimated sub-trajectory $\widetilde{BC'}$ by θ using equation (6) into \widetilde{BC} . It should be noted that the heading of a sub-trajectory is the direction pointing from the start point to the end point of this trajectory.

$$\theta = \begin{cases} heading_{BC'} - (heading_{AB} - 90^\circ) & (\text{counterclockwise}) \\ heading_{BC'} - (heading_{AB} + 90^\circ) & (\text{clockwise}) \end{cases} \quad (4)$$

$$\theta = heading_{BC'} - (heading_{AB} - 180^\circ) \quad (5)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x - x_o \\ y - y_o \end{bmatrix} + \begin{bmatrix} x_o \\ y_o \end{bmatrix} \quad (6)$$

where (x_o, y_o) is the coordinate of the rotation center (the coordinate of point B in this case). (x, y) is the coordinate of a point in the estimated sub-trajectory $\widetilde{BC'}$ and (x', y') are the rotated coordinates. The trick of this process is that we do not need to directly specify a sharply 90-degree turning but adjust the entire trajectories before and after it. This not only corrects the error that happens at the corner, but also allows the heading error accumulated during the normal walking to be removed.

The essence of our proposed method is first to split the path into sub-trajectories based on recognition of human activities and where and when they take place; second, to use the heading of the sub-trajectory of the previous activity to optimize the heading of the sub-trajectory of the current activity through coordinate rotation. There are two reasons we used 90-degree and 180-degree turning to optimize the heading. Firstly, 90-degree and 180-degree turns fit the characteristics of typical indoor layout structures of buildings. The junction between corridors is a 90-degree angle. Hence, our method can use this feature to optimize the heading. Secondly, the turning points with large-angle turning (e.g., 90-degree and 180-degree turnings) have a noticeable heading change and they can be accurately determined. As the headings of each sub-trajectory are calculated by using turning points, the turning points determine the rotation angle, θ . Therefore, 90-degree and 180-degree turnings can improve the performance of our heading optimization.

5. Positioning Experiment and Assessment

We carried out a comprehensive experiment in a multi-floor indoor building environment to demonstrate the performance of our adaptive human activity-aided PDR (HAA-PDR) system. The experiment was performed along a pre-arranged route so that all discussed human activities were performed. Detailed statistical analysis and discussions follow presentation of the method and results.

5.1. Experimental Methodology

Our experiment was conducted on the 6th, 7th, and 8th floors of the Roberts Building, Department of Electronic and Electrical Engineering, Faculty of Engineering Science, University College London (UCL), UK. It represents a typical multi-floor work/office environment. All experiments were conducted knowing the initial position coordinates and heading direction but with no further knowledge of the map of the building floors apart from the assumptions already mentioned such as 90 degree turns and straight corridors. Two young male experimenters of differing heights and weights used two smartphones (Nexus 5 and Google Pixel 2) and swapped them and repeated the trajectory. Each experimenter held a smartphone in their left hand to move along the designed route.

At each time of the experiment, an experimenter moved from the 8th floor to the 6th floor or from the 6th floor to the 8th floor. The length of the route is about 210 m. When one smartphone is used to collect data, the other one is used to film the movement of the experimenter to record the ground truth trajectory. Along the designed route, we set reference points at the turning points, doors' locations, and locations that start ascending/descending and finish ascending/descending, because these locations are important landmarks. The coordinates of these reference points were measured by using a laser range-finder (ranging accuracy is ± 1.5 mm). When the experimenter reached a reference point, a timestamp tag was manually created for the future accuracy evaluation. Note that all reference points were only used for assessment but not for automatic calculation of the trajectory. No knowledge of the map or trajectory was taken into account in the automatic calculation of the trajectories.

There were 52 reference points in total. We record the timestamps when the experimenter passes these reference points, as $[(t_1^{gt}, x_1^{gt}, y_1^{gt}), (t_2^{gt}, x_2^{gt}, y_2^{gt}) \dots, (t_{52}^{gt}, x_{52}^{gt}, y_{52}^{gt})]$. After the route had been estimated by PDR, we chose another 52 points in the estimated route according to the timestamps of reference points, as $[(t_1^{pdr}, x_1^{pdr}, y_1^{pdr}), \dots, (t_{52}^{pdr}, x_{52}^{pdr}, y_{52}^{pdr})]$. Then, we calculated the Euclidean distances between each pair of reference and estimated points as the positioning errors. This assessment metric is defined as the positioning error accumulation over time (PEA-T):

$$PEA-T = \sum_{i=1}^{52} \sqrt{(x_i^{gt} - x_i^{pdr})^2 + (y_i^{gt} - y_i^{pdr})^2} \quad I = 1, 2, \dots, 52 \quad (7)$$

5.2. System Performance Assessment

(A) Abnormal walking recognition test

Firstly, we qualitatively analyze the effects of lateral walking and stationary stepping on positioning performance. As shown in Figure 16, on the 8th floor (the size of the experimental area is about $41 \times 37 \times 9$ m³), the experimenter stands stationary for a while at point A, then performs normal walking until point B, then starts rightward lateral walking to point C, and finally performs leftward lateral walking to point B. This process is displayed as a crossed red line (the ground truth), and different activities are in different colors. Comparing the thin and dotted line, it is noticeable that the positioning accuracy can degrade sharply when the recognition of lateral walking and stationary stepping is not used. Thus, the effects of these two activities on the positioning are very significant.

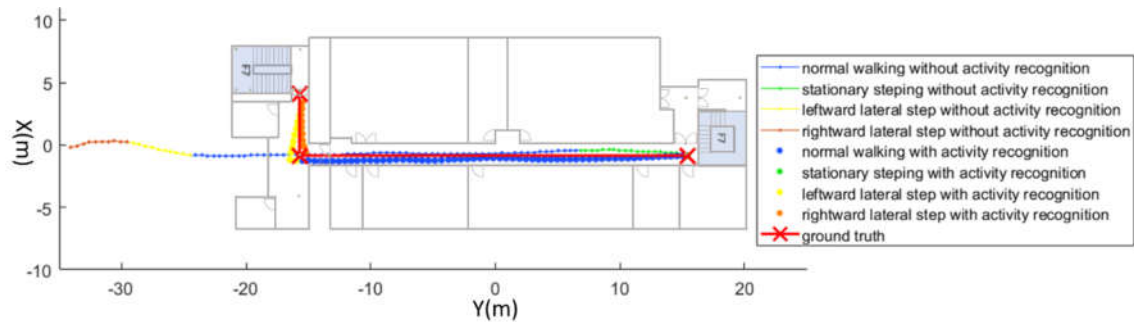


Figure 16. The positioning performance of lateral walking and stationary stepping on the 8th floor.

(B) Experiment over three floors along a designed route

Here, we discuss the performance of the designed routes. We take one experiment from multiple repetitions to show the performance of our system. In this experiment, a person walks from the 8th floor to the 6th floor. Altitude changes only happen during ascending/descending stairs. Note that lifts can also be used to change floors, but this should be clearly and simply detectable by using the barometer sensor, and hence was not investigated in this study. In particular, we included all activities discussed before except for lateral walking and stationary stepping, as their effects on PDR performance were already analyzed. Thus, we could focus more on the effects of other activities.

For activity recognition, the recognition accuracy of stationary walking and ascending/descending stairs is 98.10%, which is a little lower than that presented in Table 4 (98.44%). The reason should be the differences between the training dataset and the test dataset. In addition, fourteen door openings and 23 turns were all successfully recognized. In Figure 17a, we use different types of lines to display the trajectories of the above six activities. The cyan starred line is the stationary. The grey dotted line is the estimated trajectory of normal walking. The green starred line is the estimated trajectory of ascending/descending. The blue squared line is the estimated trajectory of door opening. The red circled line is the estimated trajectory of turning. Note that door opening and turning interfere with each other if these two activities happen very close to each other. Therefore, we deliberately walked in a way to keep each of our human activities separate from one another. In Figure 17b, we plot four strategies of PDR calculation and the ground truth to assess the performance of algorithms we proposed. All strategies are applied with the parameter optimization based on the heading-steady activities (as described in Section 4.1) for step counting and step length estimation. For the baseline PDR positioning, we do not use any trajectory optimization based on door opening or turning. However, it is dependent on HAR for the steady-head activities. For comparison, the optimization method only using door opening is called door-opening-based optimization, and the optimization method only using turning is called turning-based optimization. Our HAA-PDR system combines the door-opening- and turning-based trajectory optimization.

We look at positioning results of the above four methods from a 3D perspective, as shown in Figure 17b. The yellow triangle line (the result of baseline) deviates far from the ground truth (the red vertical-lined line) because of the positioning error accumulation mentioned earlier. The green plus line (result of door-opening-based optimization) is closer to the ground truth than the baseline. It shows that the process of door-opening-based optimization can improve the positioning accuracy of the whole route. The blue crossed line (the result of turning-based optimization) is noticeably better than the baseline, as the heading error is improved by using the turning information. Comparing results of turning-based optimization and door-opening-based optimization, heading improvement using turning information plays a key role in mitigating PDR error accumulation. Then, the results of HAA-PDR (the orange dotted line) and turning-based optimization are close to the ground truth.

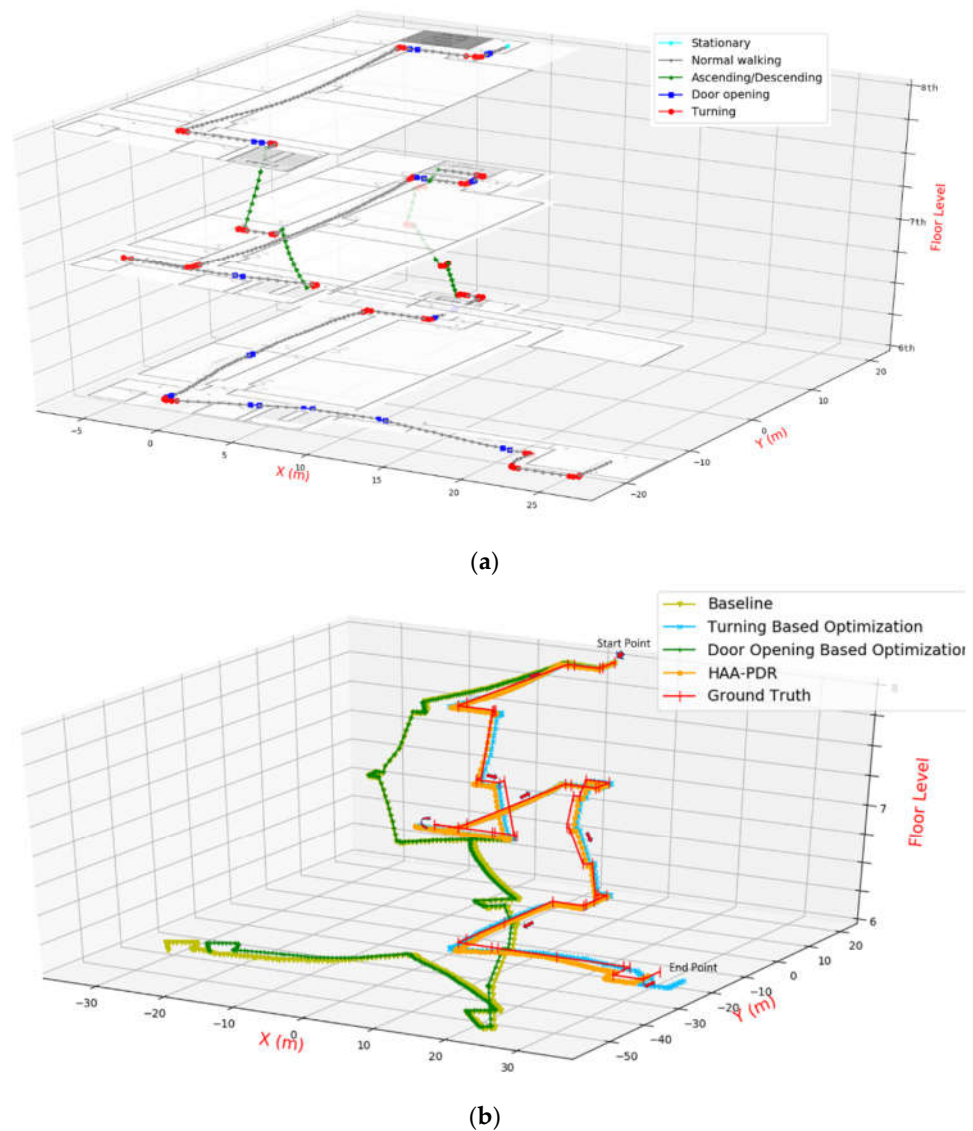
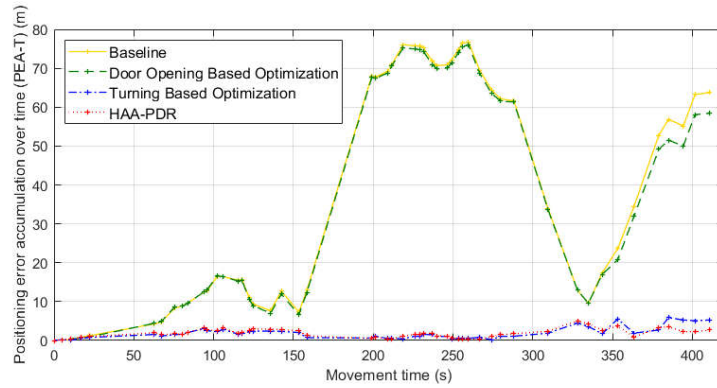


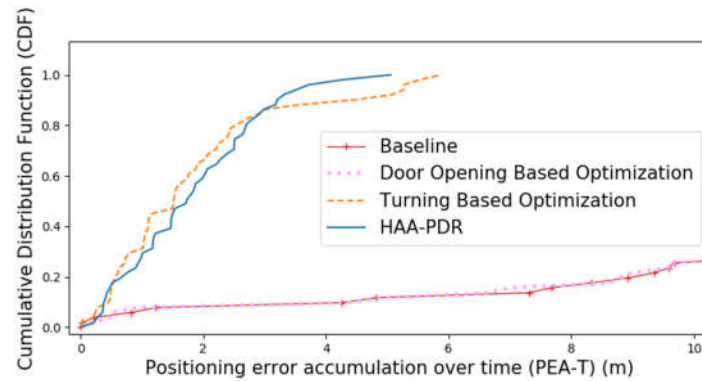
Figure 17. The results of path reconstruction at different floor levels. We also evenly interpolate the height according to the number of steps in each staircase. (a) Different activities along the path estimated by our HAA-PDR. (b) The complete paths from 3D perspective; the vertical red lines mark the places of the reference points for assessment along the ground truth trajectory.

To quantitatively analyze the performance of above four types of processing strategies, the PEA-T measured at the reference points is displayed in Figure 18a. The X-axis is the elapsed time in unite of second that the experimenter moves along the route. The Y-axis is the value of PEA-T. The marks in these PEA-T lines represent the positioning errors of each reference point. The start point of four PEA-T lines are the same, 0. Then the PEA-T of the baseline becomes higher than the other three strategies. From Table 9, the baseline's mean PEA-T is 37.73 m with a standard deviation (STD) of 29.46 m. The PEA-T of door-opening-based optimization is slightly lower than the baseline. Its mean and STD PEA-T are 36.82 and 28.95 m, respectively, which proves that the door-opening-based optimization can slightly improve the positioning performance. The PEA-T of the turning-based optimization is noticeably lower than those of the baseline and door-opening-based optimization, of which the mean and STD are 1.92 and 1.49 m, respectively. In addition,

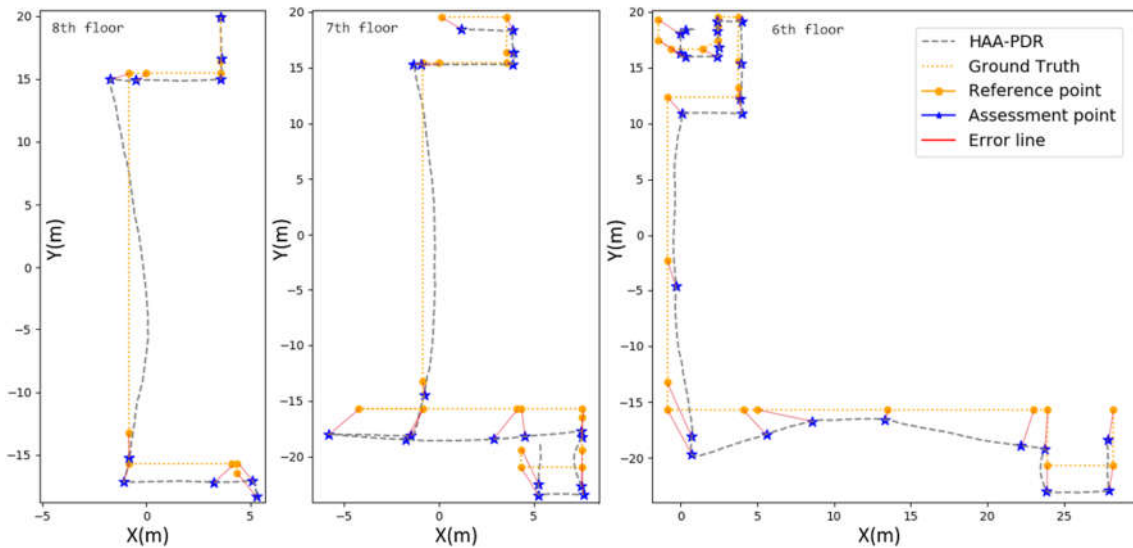
the PEA-T of HAA-PDR system is the smallest, which means it has best performance in this experiment, with a mean of 1.79 m and STD of 1.13 m. Finally, the error at the end of the trajectory of the HAA-PDR system is 2.68 m, which is far more accurate than all other methods.



(a)



(b)



(c)

Figure 18. (a) The positioning error accumulation over time (PEA-T) at each reference point for Figure 18. b. The baseline and door-opening-based optimization have very high accumulated error, PEA-T, so we show only the CDF error up to an error of 10 m. Obviously, the CDF error of the baseline and door-opening-based optimization is also very high (both can

be seen in the values of 85% and 95% CDF error listed in Table 9). As for the CDF error of turning-based optimization and HAA-PDR, we can see that these two methods' CDF is very close at first. After the CDF reaches about 80%, HAA-PDR gradually outperforms the turning-based optimization. Finally, our HAA-PDR achieves an accuracy of 2.98 m in 85% CDF error and 3.72 m in 95% CDF error, which both surpass all other methods. To better visualize the positioning error of our HAA-PDR system, we display the PEA-T details of assessment points in Figure 18c. The so-called assessment points are defined as the estimation of reference points by our HAA-PDR system. The gray dashed line and the yellow dotted line represent the HAA-PDR and ground truth, respectively. The yellow points in the line of ground truth are reference points. The blue stars in the line of HAA-PDR are the assessment points. The red short connection lines between each pair of reference and assessment points represent the PEA-T.

From the mean, STD, and CDF of PEA-T in Table 9, we can see that door-opening- and turning-based trajectory optimization are able to improve the positioning performance. Compared to the baseline, the mean accuracy improvement using the turning-based optimization is 36.82 m (approximately 20 times); the mean accuracy improved by the door-opening-based optimization is only 0.91 m. Thus, the turning-based optimization gives a more noticeable improvement than the door-opening-based optimization in this experiment. Both door opening and turning-based trajectory optimization can effectively improve the positioning performance, and clearly the HAA-PDR is the best, as it combines both and performs the best in all assessment categories.

6. Conclusions

In this paper, we proposed and demonstrated a new adaptive human activity-aided pedestrian dead reckoning (HAA-PDR) positioning system to not only improve the robustness but also mitigate the error accumulation of traditional PDR systems. We defined six steady-heading activities (stationary, walking, ascending, descending, stationary stepping, and lateral walking) and two non-steady-heading activities (door opening and turning) to optimize PDR performance. A novel hierarchical combination of SVM and DT is proposed to recognize steady-heading activities. We proposed and demonstrated a novel autoencoder-based deep neural network and a heading range-based method to respectively recognize door opening and turning. The most accurate HAR methods are adaptive with different activities to achieve an over 98.44% accuracy of recognition. After the human activity recognition, steady-heading activities were used to set the parameters for step counting and step length estimation methods for each activity separately to improve the robustness. After that, we broke the trajectory into small segments based on activities. Then, for each activity trace, especially for non-steady-heading activities, we designed two optimization methods specifically to optimize the PDR trajectory and then reconstructed the whole PDR trajectory. The result of a series of experiments conducted in a multi-floor building demonstrates the performance of our HAA-PDR system, which is superior to the conventional PDR system.

It is a challenging problem to use human activity recognition to improve the accuracy of a PDR positioning system. First, there are many types of human locomotion activities that affect the accuracy of the PDR system, so deciding which activities to focus on is problematic. Another issue is that the same locomotion activity may be performed in different ways by the same person (i.e., as they get tired) and by different people so they vary in time and space. Our work shows an interesting example that utilizes multiple activities to improve PDR performance. However, it was a limitation that our system was built and tested by only two experimenters. In future work, we will consider more phone poses, more locomotion activities (e.g., using a lift, running because we are in a rush), and include more than two experimenters having a diverse range of physical characteristics for training and testing to further improve the positioning performance and hence to construct a more robust and environment-adaptive PDR system.

Author Contributions: Conceptualization, B.W and C.M; Data curation, B.W and C.M; Formal analysis, B.W and C.M; Investigation, B.W and C.M; Methodology, B.W and C.M; Software, B.W; Supervision, S.P and D.R.S; Validation, B.W and C.M; Visualization, B.W; Writing – original draft, B.W

and C.M; Writing – review & editing, C.M, S.P and D.R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by a Ph.D. scholarship funded jointly by the China Scholarship Council (CSC) and QMUL (for Bang Wu).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Zhang, W.; Yu, K.; Wang, W.; Li, X. A self-adaptive ap selection algorithm based on multi-objective optimization for indoor wifi positioning. *IEEE Internet Things J.* **2020**, *8*, 1406–1416.
- Li, Z.; Liu, C.; Gao, J.; Li, X. An improved wifi/pdr integrated system using an adaptive and robust filter for indoor localization. *ISPRS Int. J. Geo Inf.* **2016**, *5*, 224.
- Ma, C.; Wu, B.; Poslad, S.; Selviah, D.R. Wi-Fi RTT ranging performance characterization and positioning system design. *IEEE Trans. Mob. Comput.* **2020**, *21*, 8479–8490.
- Ssekidde, P.; Steven Eyobu, O.; Han, D.S.; Oyana, T.J. Augmented CWT features for deep learning-based indoor localization using WiFi RSSI data. *Appl. Sci.* **2021**, *11*, 1806, doi:10.3390/app11041806.
- Poulose, A.; Han, D.S. Hybrid deep learning model based indoor positioning using Wi-Fi RSSI heat maps for autonomous applications. *Electronics* **2021**, *10*, 2, doi:10.3390/electronics10010002.
- Chen, P.; Kuang, Y.; Chen, X. A UWB/improved PDR integration algorithm applied to dynamic indoor positioning for pedestrians. *Sensors* **2017**, *17*, 2065.
- Feng, D.; Wang, C.; He, C.; Zhuang, Y.; Xia, X.-G. Kalman-filter-based integration of IMU and UWB for high-accuracy indoor positioning and navigation. *IEEE Internet Things J.* **2020**, *7*, 3133–3146.
- Poulose, A.; Han, D. UWB indoor localization using deep learning LSTM Networks. *Appl. Sci.* **2020**, *10*, 6290, doi:10.3390/app10186290.
- Li, X.; Wang, J.; Liu, C. A Bluetooth/PDR integration algorithm for an indoor positioning system. *Sensors* **2015**, *15*, 24862–24885.
- Bai, L.; Ciravegna, F.; Bond, R.; Mulvenna, M. A low cost indoor positioning system using bluetooth low energy. *IEEE Access* **2020**, *8*, 136858–136871, doi:10.1109/ACCESS.2020.3012342.
- Potorti, F.; Park, S.; Crivello, A.; Palumbo, F.; Girolami, M.; Barsocchi, P.; Lee, S.; Torres-Sospedra, J.; Jimenez, A.R.; Pérez-Navarro, A. The IPIN 2019 indoor localisation competition-description and results. *IEEE Access* **2020**, *8*, 206674–206718.
- Yousif, K.; Bab-Hadiashar, A.; Hoseinnezhad, R. An overview to visual odometry and visual SLAM: Applications to mobile robotics. *Intell. Ind. Syst.* **2015**, *1*, 289–311.
- Poulose, A.; Han, D.S. Hybrid indoor localization using IMU sensors and smartphone camera. *Sensors* **2019**, *19*, 5084, doi:10.3390/s19235084.
- Plets, D.; Almadani, Y.; Bastiaens, S.; Ijaz, M.; Martens, L.; Joseph, W. Efficient 3D trilateration algorithm for visible light positioning. *J. Opt.* **2019**, *21*, 05LT01, doi:10.1088/2040-8986/ab1389.
- Lachhani, K.; Duan, J.; Baghsiahi, H.; Willman, E.; Selviah, D.R. Correspondence rejection by trilateration for 3D point cloud registration. In Proceedings of the 2015 14th IAPR International Conference on Machine Vision Applications (MVA), Tokyo, Japan, 18–22 May 2015; IEEE: New York, NY, USA, 2015; pp. 337–340.
- Lain, J.-K.; Chen, L.-C.; Lin, S.-C. Indoor localization using k-pairwise light emitting diode image-sensor-based visible light positioning. *IEEE Photonics J.* **2018**, *10*, 1–9, doi:10.1109/JPHOT.2018.2881239.
- Chen, Y.; Liu, J.; Jaakkola, A.; Hyyppä, J.; Chen, L.; Hyyppä, H.; Jian, T.; Chen, R. Knowledge-based indoor positioning based on LiDAR aided multiple sensors system for UGVs. In Proceedings of the 2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014, Monterey, CA, USA, 5–8 May 2014; IEEE: New York, NY, USA, 2014; pp. 109–114.
- David, R.S.; Eero, W. Apparatus, Method and System for Alignment of 3D Datasets. GB2559157A, 1 August 2018.
- Xu, Y.; Shmaliy, Y.S.; Li, Y.; Chen, X.; Guo, H. Indoor INS/LiDAR-based robot localization with improved robustness using cascaded fir filter. *IEEE Access* **2019**, *7*, 34189–34197, doi:10.1109/ACCESS.2019.2903435.
- Holm, S.; Nilsen, C.-I. C. Robust ultrasonic indoor positioning using transmitter arrays. In Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation, Zurich, Switzerland, 15–17 September 2010; IEEE: New York, NY, USA, 2010; pp. 1–5.
- Li, J.; Han, G.; Zhu, C.; Sun, G. An indoor ultrasonic positioning system based on TOA for internet of things. *Mob. Inf. Syst.* **2016**, *2016*, 1–10, doi:10.1155/2016/4502867.

22. Ma, Z.; Poslad, S.; Hu, S.; Zhang, X. A fast path matching algorithm for indoor positioning systems using magnetic field measurements. In Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, CA, 8–13 October 2017; IEEE: New York, NY, USA, 2017; pp. 1–5.
23. Xu, L.; Xiong, Z.; Liu, J.; Wang, Z.; Ding, Y. A novel pedestrian dead reckoning algorithm for multi-mode recognition based on smartphones. *Remote Sens.* **2019**, *11*, 294.
24. Poulou, A.; Eyobu, O.S.; Han, D.S. An indoor position-estimation algorithm using smartphone imu sensor data. *IEEE Access* **2019**, *7*, 11165–11177, doi:10.1109/ACCESS.2019.2891942.
25. Ma, C.; Wan, C.; Chau, Y.W.; Kang, S.M.; Selviah, D.R. Subway station real-time indoor positioning system for cell phones. In Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, Japan, 18–21 September 2017; pp. 1–7, doi:10.1109/IPIN.2017.8115912.
26. Gebre-Egziabher, D.; Hayward, R.C.; Powell, J.D. A low-cost gps/inertial attitude heading reference system (ahrs) for general aviation applications. In Proceedings of the IEEE 1998 Position Location and Navigation Symposium (Cat. No. 98CH36153), Palm Springs, CA, USA, 20–23 April 1996; IEEE: New York, NY, USA, 1996; pp. 518–525.
27. Poulou, A.; Senouci, B.; Han, D.S. Performance analysis of sensor fusion techniques for heading estimation using smartphone sensors. *IEEE Sens. J.* **2019**, *19*, 12369–12380, doi:10.1109/JSEN.2019.2940071.
28. Tian, Q.; Salci, Z.; Kevin, I.; Wang, K.; Pan, Y. A multi-mode dead reckoning system for pedestrian tracking using smartphones. *IEEE Sens. J.* **2015**, *16*, 2079–2093.
29. Wang, B.; Liu, X.; Yu, B.; Jia, R.; Gan, X. Pedestrian dead reckoning based on motion mode recognition using a smartphone. *Sensors* **2018**, *18*, 1811.
30. Sheu, J.S.; Huang, G.S.; Jheng, W.C.; Hsiao, C.H. Design and implementation of a three-dimensional pedometer accumulating walking or jogging motions. In Proceedings of the 2014 International Symposium on Computer, Consumer and Control, Taichung, Taiwan, 10–12 June 2014; IEEE: New York, NY, USA, 2014; pp. 828–831.
31. Jang, H.-J.; Kim, J.W.; Hwang, D.-H. Robust step detection method for pedestrian navigation systems. *Electron. Lett.* **2007**, *43*, 749–751.
32. Rai, A.; Chintalapudi, K.K.; Padmanabhan, V.N.; Sen, R. Zee: Zero-effort crowdsourcing for indoor localization. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, August 22–26 2012; ACM: New York, NY, USA, 2012; pp. 293–304.
33. Gu, F.; Khoshelham, K.; Shang, J.; Yu, F.; Wei, Z. Robust and accurate smartphone-based step counting for indoor localization. *IEEE Sens. J.* **2017**, *17*, 3453–3460.
34. Kim, J.W.; Jang, H.J.; Hwang, D.-H.; Park, C. A step, stride and heading determination for the pedestrian navigation system. *J. Glob. Position. Syst.* **2004**, *3*, 273–279, doi:10.5081/jgps.3.1.273.
35. Weinberg, H. Using the ADXL202 in pedometer and personal navigation applications. *Analog Devices 602 Appl. Note* **2002**, *2*, 1–6.
36. Ladetto, Q. On foot navigation: Continuous step calibration using both complementary recursive prediction and adaptive kalman filtering. In Proceedings of the ION GPS, Salt Lake City, UT, USA, 19–22 September 2000; Volume 2000, pp. 1735–1740.
37. Fan, B.; Li, Q.; Liu, T. How magnetic disturbance influences the attitude and heading in magnetic and inertial sensor-based orientation estimation. *Sensors* **2018**, *18*, 76.
38. Madgwick, S.O.H.; Harrison, A.J.L.; Vaidyanathan, R. Estimation of IMU and MARG orientation using a gradient descent algorithm. In Proceedings of the 2011 IEEE International Conference on Rehabilitation Robotics, Zurich, Switzerland, 29 June 2011; IEEE: New York, NY, USA, 2011; pp. 1–7.
39. Foxlin, E. Inertial head-tracker sensor fusion by a complementary separate-bias kalman filter. In Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium, Santa Clara, CA, USA, 30 March–3 April 1996; IEEE: New York, NY, USA, 1996; pp. 185–194.
40. Mahony, R.; Hamel, T.; Pflimlin, J.-M. Nonlinear complementary filters on the special orthogonal group. *IEEE Trans. Autom. Control* **2008**, *53*, 1203–1217.
41. Martinelli, A.; Gao, H.; Groves, P.D.; Morosi, S. Probabilistic context-aware step length estimation for pedestrian dead reckoning. *IEEE Sens. J.* **2017**, *18*, 1600–1611.
42. Elhoushi, M.; Georgy, J.; Noureldin, A.; Korenberg, M.J. Motion mode recognition for indoor pedestrian navigation using portable devices. *IEEE Trans. Instrum. Meas.* **2015**, *65*, 208–221.
43. Susi, M.; Renaudin, V.; Lachapelle, G. Motion mode recognition and step detection algorithms for mobile phone users. *Sensors* **2013**, *13*, 1539–1562.
44. Shin, B.; Kim, C.; Kim, J.; Lee, S.; Kee, C.; Kim, H.S.; Lee, T. Motion recognition-based 3D pedestrian navigation system using smartphone. *IEEE Sens. J.* **2016**, *16*, 6977–6989.
45. Elhoushi, M.; Georgy, J.; Wahdan, A.; Korenberg, M.; Noureldin, A. Using portable device sensors to recognize height changing modes of motion. In Proceedings of the 2014 IEEE International Instrumentation and Measurement Technology Conference (I2mtc) Proceedings, Montevideo, Uruguay, 11–14 May 2014; IEEE: New York, NY, USA, 2014; pp. 477–481.
46. Gu, F.; Kealy, A.; Khoshelham, K.; Shang, J. User-independent motion state recognition using smartphone sensors. *Sensors* **2015**, *15*, 30636–30652.
47. Pei, L.; Liu, J.; Guinness, R.; Chen, Y.; Kuusniemi, H.; Chen, R. Using LS-SVM based motion recognition for smartphone indoor wireless positioning. *Sensors* **2012**, *12*, 6155–6175.

48. Zebin, T.; Scully, P.J.; Ozanyan, K.B. Human activity recognition with inertial sensors using a deep learning approach. In Proceedings of the 2016 IEEE SENSORS, Orlando, FL, USA, 30 October–3 November 2016; IEEE: New York, NY, USA, 2016; pp. 1–3, doi:10.1109/ICSENS.2016.7808590.
49. Xia, K.; Huang, J.; Wang, H. LSTM-CNN architecture for human activity recognition. *IEEE Access* **2020**, *8*, 56855–56866, doi:10.1109/ACCESS.2020.2982225.
50. Tufek, N.; Yalcin, M.; Altintas, M.; Kalaoglu, F.; Li, Y.; Bahadir, S.K. Human action recognition using deep learning methods on limited sensory data. *IEEE Sens. J.* **2020**, *20*, 3101–3112, doi:10.1109/JSEN.2019.2956901.
51. Yu, S.; Qin, L. Human activity recognition with smartphone inertial sensors using bidir-LSTM networks. In Proceedings of the 2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Huhhot, China, 14–16 September 2018; IEEE: New York, NY, USA, 2018; pp. 219–224, doi:10.1109/ICMCCE.2018.00052.
52. Rhea, C.K.; Wittstein, M.W. Characteristics of stride behavior during treadmill walking and stationary stepping. *J. Appl. Biomech.* **2014**, *30*, 534–541.
53. Wang, X.; Jiang, M.; Guo, Z.; Hu, N.; Sun, Z.; Liu, J. An indoor positioning method for smartphones using landmarks and PDR. *Sensors* **2016**, *16*, 2135.
54. Ghosh, A.; Riccardi, G. Recognizing human activities from smartphone sensor signals. In Proceedings of the 22nd ACM International Conference on Multimedia; MM '14, Orlando, FL, USA, November 2014; Association for Computing Machinery: Orlando, FL, USA, 2014; pp. 865–868, doi:10.1145/2647868.2655034.
55. He, W.; Guo, Y.; Gao, C.; Li, X. Recognition of human activities with wearable sensors. *EURASIP J. Adv. Signal Process.* **2012**, *2012*, 1–3.
56. Cornacchia, M.; Ozcan, K.; Zheng, Y.; Velipasalar, S. A survey on activity detection and classification using wearable sensors. *IEEE Sens. J.* **2016**, *17*, 386–403.
57. Scholkopf, B.; Sung, K.-K.; Burges, C.J.; Girosi, F.; Niyogi, P.; Poggio, T.; Vapnik, V. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Trans. Signal Process.* **1997**, *45*, 2758–2765.
58. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359, doi:10.1109/TKDE.2009.191.
59. Moreno-Torres, J.G.; Sáez, J.A.; Herrera, F. Study on the impact of partition-induced dataset shift on k -fold cross-validation. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 1304–1312.
60. Ng, A. Sparse autoencoder. *CS294A Lect. Notes* **2011**, *72*, 1–19.
61. Selviah, D.R.; Midwinter, J.E. Extension of the hamming neural network to a multilayer architecture for optical implementation. In Proceedings of the 1989 First IEE International Conference on Artificial Neural Networks, (Conf. Publ. No. 313), London, UK, 16–18 October 1989; IET: Stevenage, UK, 1989; pp. 280–283.
62. Selviah, D.R.; Midwinter, J.E.; Rivers, A.W.; Lung, K.W. Correlating matched-filter model for analysis and optimisation of neural networks. In *IEE Proceedings F (Radar and Signal Processing)*; IET: Stevenage, UK, 1989; Volume 136, pp. 143–148.
63. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *ArXiv Prepr.* **2014**, ArXiv14126980.
64. Lu, Z.; He, R.; Li, J. Research on Online comment negative emotion warning based on machine learning from the perspective of big data. *DEStech Trans. Comput. Sci. Eng.* **2019**, doi:10.12783/dtcse/cscme2019/32540.
65. Zhou, B.; Li, Q.; Mao, Q.; Tu, W.; Zhang, X.; Chen, L. ALIMC: Activity landmark-based indoor mapping via crowdsourcing. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2774–2785.
66. Fang, L.; Antsaklis, P.J.; Montestruque, L.A.; McMickell, M.B.; Lemmon, M.; Sun, Y.; Fang, H.; Koutroulis, I.; Haenggi, M.; Xie, M. Design of a wireless assisted pedestrian dead reckoning system—the navmote experience. *IEEE Trans. Instrum. Meas.* **2005**, *54*, 2342–2358.
67. Zhang, M.; Shen, W.; Yao, Z.; Zhu, J. Multiple information fusion indoor location algorithm based on wifi and improved PDR. In Proceedings of the 2016 35th Chinese Control Conference (CCC), Chengdu, China, 27–29 July 2016; IEEE: New York, NY, USA, 2016; pp. 5086–5092.
68. Oubre, B.; Daneault, J.-F.; Boyer, K.; Kim, J.H.; Jasim, M.; Bonato, P.; Lee, S.I. A simple low-cost wearable sensor for long-term ambulatory monitoring of knee joint kinematics. *IEEE Trans. Biomed. Eng.* **2020**, 1–1, doi:10.1109/TBME.2020.2988438.