

Article

Pedestrian Walking Distance Estimation Based on Smartphone Mode Recognition

Qu Wang ¹, Langlang Ye ², Haiyong Luo ^{2,*}, Aidong Men ¹, Fang Zhao ³ and Changhai Ou ⁴

¹ School of Information and Communication Engineering, Beijing University of Posts and Telecommunication, Beijing 100876, China; wangqu@ict.ac.cn (Q.W.); menad@bupt.edu.cn (A.M.)

² Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology Chinese Academy of Sciences, Beijing 100190, China; yelanglang@ict.ac.cn

³ School of Software Engineering, Beijing University of Posts and Telecommunication, Beijing 100876, China; zfsse@bupt.edu.cn

⁴ School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore; chou@ntu.edu.sg

* Correspondence: yhluo@ict.ac.cn

Received: 17 April 2019; Accepted: 11 May 2019; Published: 13 May 2019



Abstract: Stride length and walking distance estimation are becoming a key aspect of many applications. One of the methods of enhancing the accuracy of pedestrian dead reckoning is to accurately estimate the stride length of pedestrians. Existing stride length estimation (SLE) algorithms present good performance in the cases of walking at normal speed and the fixed smartphone mode (handheld). The mode represents a specific state of the carried smartphone. The error of existing SLE algorithms increases in complex scenes with many mode changes. Considering that stride length estimation is very sensitive to smartphone modes, this paper focused on combining smartphone mode recognition and stride length estimation to provide an accurate walking distance estimation. We combined multiple classification models to recognize five smartphone modes (calling, handheld, pocket, armband, swing). In addition to using a combination of time-domain and frequency-domain features of smartphone built-in accelerometers and gyroscopes during the stride interval, we constructed higher-order features based on the acknowledged studies (Kim, Scarlett, and Weinberg) to model stride length using the regression model of machine learning. In the offline phase, we trained the corresponding stride length estimation model for each mode. In the online prediction stage, we called the corresponding stride length estimation model according to the smartphone mode of a pedestrian. To train and evaluate the performance of our SLE, a dataset with smartphone mode, actual stride length, and total walking distance were collected. We conducted extensive and elaborate experiments to verify the performance of the proposed algorithm and compare it with the state-of-the-art SLE algorithms. Experimental results demonstrated that the proposed walking distance estimation method achieved significant accuracy improvement over existing individual approaches when a pedestrian was walking in both indoor and outdoor complex environments with multiple mode changes.

Keywords: indoor positioning; machine learning; pedestrian dead reckoning; stride length estimation; smartphone mode recognition

1. Introduction

Applications that attempt to track pedestrian motion level (walking distance) for health purposes require an accurate step detection and stride length estimation (SLE) technique [1]. Walking distance is used to assess the physical activity level of the user, which helps provide feedback and motivate a more

active lifestyle [2,3] Another type of application based on walking distance is navigation applications. Among various indoor localization methods, pedestrian dead reckoning (PDR) [4] has become a mainstream and practical method, because PDR does not require any infrastructure. In addition to the general applications, involving asset and personnel tracking, health monitoring, precision advertising, and location-specific push notifications, PDR is available for emergency scenarios, such as anti-terrorism action, emergency rescue, and exploration missions. Furthermore, smartphone-based PDR mainly benefits from the extensive use of smartphones—pedestrians always carry smartphones that have integrated inertial sensors. Stride length estimation is a key component of PDR, the accuracy of which will directly affect the performance of PDR systems. Therefore, in addition to providing more accurate motion level estimation, precise stride length estimation based on built-in smartphone inertial sensors enhances positioning accuracy of PDR. Most visible light positioning [5,6], Wi-Fi positioning [7–9], and magnetic positioning [10–12] critically depend on PDR. Hence, motion level estimation based on smartphones contributes to assisting and supporting patients undergoing health rehabilitation and treatment, activity monitoring of daily living, navigation, and numerous other applications [13].

The methods for estimating pedestrian step length are summarized as two categories: the first is direct methods, based on the integration of acceleration; the second is indirect methods that leverage a model or assumption to compute step length. The double integration of the acceleration component in the forward direction is the best method to compute the stride length of pedestrians because it does not rely on any model or assumption, and does not require training phases or individual information (leg length, height, weight) [14]. Kouroggi et al. [15] leveraged the correlation between vertical acceleration and walking velocity to estimate walking speed, and calculated stride length by multiplying walking speed with step interval. However, the non-negligible bias and noise of the accelerometers and gyroscopes resulted in the distance error growing boundlessly and cubically in time [14]. Moreover, it is difficult to obtain the acceleration component in the forward direction from the sensor's measurements, as well as constantly maintaining the sensor heading parallel to the pedestrian's walking direction [16]. Additionally, low-cost smartphone sensors are not reliable and accurate enough to estimate the stride length of a pedestrian by double integrating the acceleration [17]. Developing a step length estimation algorithm using MEMS (micro-electro-mechanical systems) sensors is recognized as a difficult problem.

Considerable research based on models or assumptions has been conducted to improve the accuracy of SLE, and summarized as empirical relationships [18,19], biomechanical models [18,20,21], linear models [22], nonlinear models [23–25], regression-based [22,26], and neural networks [27–30]. One of the most renowned SLE algorithms was presented by Weinberg [23]. To estimate the walk distance, he leveraged the range of the vertical acceleration values during each step, according to Equation (1).

$$L = k \cdot \sqrt[4]{a_{\max} - a_{\min}} \quad (1)$$

where a_{\max} and a_{\min} denote the maximum and minimum acceleration values on the Z-axis in each stride, respectively. k represents the calibration coefficient, which is obtained from the ratio of the actual distance and the estimated distance.

As shown in Equation (2), Kim et al. [24] developed an empirical method, based on the average of the acceleration magnitude in each stride during walking, to calculate movement distance.

$$L = k \cdot \sqrt[3]{\frac{\sum_{i=1}^N |a_i|}{N}} \quad (2)$$

where a_i represents the measured acceleration value of the i th sample in each step, and N represents the number of samples corresponding to each step. k is the calibration coefficient.

To estimate the travel distance of a pedestrian accurately, Ladetto et al. [22] leveraged the linear relationship between step length and frequency and the local variance of acceleration to calculate the motion distance with the following equation:

$$L = \alpha \cdot f + \beta \cdot v + \gamma \quad (3)$$

where f is the step frequency, which represents the reciprocal of one stride interval, v is the acceleration variance during the interval of one step, α and β denote the weighting factors of step frequency and acceleration variance, respectively, and γ represents a constant that is used to fit the relationship between the actual distance and the estimated distance.

Kang et al. [31] simultaneously measured the inertial sensor and global positioning system (GPS) position while walking outdoors with a reliable GPS fix, and regarded the velocity from the GPS as labels to train a hybrid multiscale convolutional and recurrent neural network model. After that, Kang leveraged the prediction velocity and moving time to estimate the traveled distance. However, it is challenging to obtain accurate labels, since GPS contains a positional error. Zhu et al. [32] measured the duration of the swing phase in each gait cycle by accelerometer and gyroscope, and then combined the acceleration information during the swing phase to obtain the step length. Xing et al. [29] proposed a stride length estimation algorithm based on a back propagation artificial neural network, using a consumer-grade inertial measurement unit. To eliminate the effect of the accelerometer bias and the acceleration of gravity, Cho et al. [28] utilized a neural network method for step length estimation. Martinelli et al. [33] proposed a weighted, context-based step length estimation (WC-SLE) algorithm, in which the step lengths computed for different pedestrian contexts were weighted by the context probabilities. Diaz et al. [34] and Diaz [35] leveraged an inertial sensor mounted on the thigh and the variation amplitude of the leg's pitch as a predictor to build a linear regression model. To reduce overfitting, Zihajehzadeh and Park [36,37] used lasso regression to fit the linear model by minimizing a penalized version of the least squares loss function. However, these methods required the user to wear a special device in a specific position on the body. In our previous work [38], a stride length estimation method based on long short-term memory (LSTM) and denoising autoencoders (DAE), termed Tapeline, was proposed. Tapeline [38] first leveraged a LSTM network to excavate the temporal dependencies and extract significant features vectors from noisy inertial sensor measurements. Afterwards, denoising autoencoders were adopted to sanitize the inherent noise and obtain denoised feature vectors automatically. Finally, a regression module was employed to map the denoised feature vectors to the resulting stride length. Tapeline achieved superior performance, with a stride length error rate of 4.63% and a walking distance error rate of 1.43%. However, the LSTM network and denoising processes result in a large number of computational overheads. The most significant drawback of the Tapeline was that pedestrians should hold their phone horizontally with their hand in front of their chest.

Stride length and walking distance estimation from smartphones' inertial sensors are challenging because of the various walking patterns and smartphone carrying methods. These SLE algorithms perform well in the case of walking at normal speed under fixed mode. Unfortunately, the pedestrian may walk arbitrarily in different directions and may stop from time to time. Moreover, real paths always including turns, sidesteps, stairs, variations in speed, or various actions performed by the subject, which will result in unacceptable SLE accuracy. Eventually, these algorithms' performances are shown to be highly sensitive to the carrying position of a smartphone (smartphone modes) on the user's body. As shown in Table 1, there are significant differences in the mean and standard deviation of acceleration and gyroscope collected under different carrying position.

Table 1. The mean and standard deviation of acceleration and gyroscope collected under different smartphone modes.

Modes	Handheld		Calling		Swing-Hand		Arm-Hand		Trouser Pocket	
	acc (m/s ²)	gyro (rad/s)								
Mean	9.59	0.15	9.79	0.53	10.33	0.99	9.84	0.76	11.02	1.75
STD	0.65	0.14	1.03	0.95	2.38	1.25	1.61	0.55	4.19	1.01

To overcome the shortcomings above of previous works, we proposed a pedestrian walking distance estimation method independent of smartphone mode. We first recognized the smartphone modes automatically by a position classifier, and then selected the most suitable stride length model for each smartphone mode. To our knowledge, we are the first to estimate the stride length and walking distance based on smartphone mode recognition, which aims to mitigate the impact of different smartphone carrying modes, thus significantly improving its stride length estimation accuracy. The key contributions of our study are as follows:

- In addition to the combination of time-domain and frequency-domain features of accelerometers and gyroscopes during the stride interval, we also built higher-order features based on the acknowledged studies to model the stride length.
- We developed a computational lightweight smartphone mode recognition method that performed accurately using inertial signals. The proposed smartphone mode recognition method achieved a recognition accuracy of 98.82% by using a two layer stacking model.
- We fused multiple regression predictions from different regression models in machine learning using a stacking regression model, so that we obtained an optimal stride length estimation accuracy with an error rate of 3.30%, dependent only on the embedded smartphone inertial sensor data.
- We established a benchmark dataset with ground truth from a FM-INS (foot-mounted inertial navigation system, x-IMU [39] from x-io technologies) module for step counting, smartphone mode recognition, and stride length or walking distance estimation. We trained different stride length models for common smartphone modes and estimated the walking distance of the pedestrian by automatically recognizing the smartphone modes and selecting the most suitable stride length model. The proposed method achieved a superior performance to traditional methods, with a walking distance error rate of 2.62%.

The rest of the paper is organized as follows: In Section 2, we describe the benchmark dataset and the feature extraction, then detail the solution of smartphone mode recognition and stride length estimation. In Section 3, numerical results and performance comparison are presented in detail. In Section 4, we provide a discussion and conclusion that summarizes the importance and limitations of our proposed work, and give suggestions for future research.

2. Walking Distance Estimation Based on Smartphone Mode Recognition

Figure 1 depicts the architecture of the proposed pedestrian walking distance estimation method, which includes three main stages: classifier training for smartphone modes recognition, stride length model training, and online stride length estimation. After a low pass filter, the time- and frequency-domain features of built-in smartphone inertial sensors during the stride interval was extracted in three stages. To train the stride length models and evaluate the stride length estimation results, we leveraged the FM-INS module to obtain the real and precise motion distance of each stride. We utilized the extracted features and the corresponding motion distance from the FM-INS module to train the stride length regression models in the offline phase. Meanwhile, a classifier model was trained to identify the carrying mode of the smartphone. During the online predicting, we integrated the extracted features, and trained classifier and stride length models to predict the stride length of each stride, as well as walking distance.

2.1. Benchmark Dataset

The lack of benchmarking datasets for pedestrian SLE makes it hard to pinpoint the differences of published methods. Existing datasets either lack the ground-truth of stride length and smartphone modes or are limited to small spaces with a single scene. To fully evaluate the performance of the proposed SLE algorithm, we constructed a benchmark dataset for natural pedestrian dead reckoning using smartphone sensors and a FM-INS module (includes a three-axis accelerometer and a three-axis gyroscope). The FM-INS module communicates with the smartphone via a BLE AIR (Bluetooth Low Energy air) interface. The gyroscope detects the angular motion, while the accelerometer detects the linear motion. The x-IMU's performance parameters are listed in Table 2. We leveraged the x-IMU module to provide the ground-truth of each stride with a motion distance error of 0.3% across the entire travel distance. As shown in Figure 2, the FM-INS module was attached to the right foot. The coordinates of the x-IMU module were defined as follows: the Y-axis along the toe direction; the Z-axis perpendicular to the foot surface, and the X-axis conforming to the right-hand rule. The motion data were captured using a Huawei Mate 9 smartphone, which comprised an accelerometer and a gyroscope, to acquire tri-axial acceleration and angular velocity signals at a sampling rate of 100 Hz. The dataset was obtained from a group of healthy adults with natural motion patterns (fast walking, normal walking, slow walking). Table 3 details the information of the subjects. During data acquisition, collectors randomly changed their smartphone mode while updating the corresponding data label (Handheld, Arm-hand, Pocket, Calling, and Swing). The distributions for the five modes are shown in Figure 3. The datasets contain more than 13.5 km and 10,145 strides of gait measurements. Figure 4 shows the distribution of real stride length in all datasets. The mean and deviation of the stride length were 1.33 m and 0.18, respectively. A total 99.5% of strides were within 1.55 m. All strides were within 1.75 m. Therefore, this paper concludes that the stride length of a pedestrian is $[0,2]$. We set the regression prediction range of pedestrian stride length between 0 and 2 m. Special cases such as jumping are not within the scope of this article. To reduce redundancy and maximize compatibility, all the data were published in JSON (JavaScript Object Notation) format. As shown in Figure 5, each stride holds nine degrees-of-freedom sensor data and the corresponding stride number, smartphone mode, stride length, and total walking distance. More detailed info about the dataset can be found in GitHub (<https://github.com/wq1989/WalkingDistanceEstimation>).

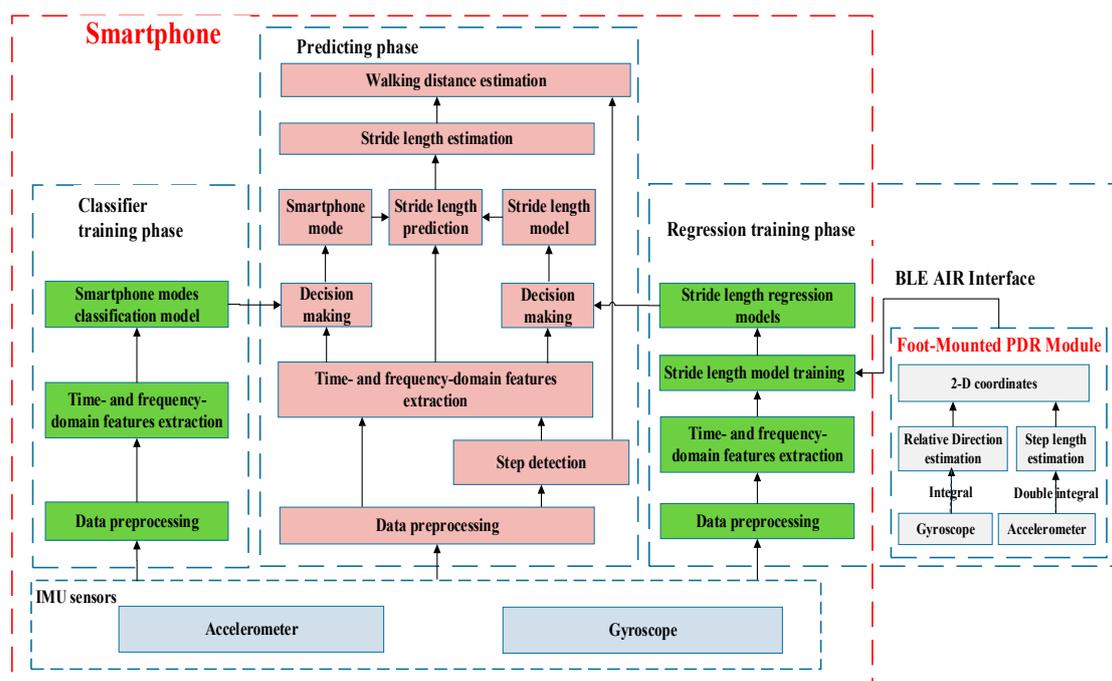


Figure 1. The system architecture of the proposed walking distance estimation method.

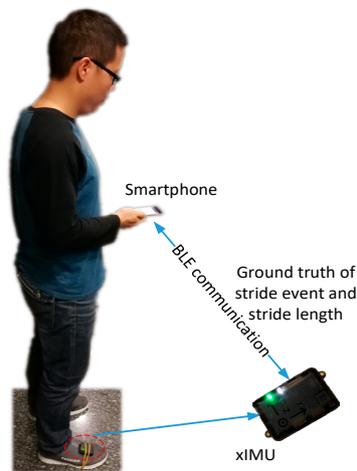


Figure 2. The system of training data collection and performance evaluation.

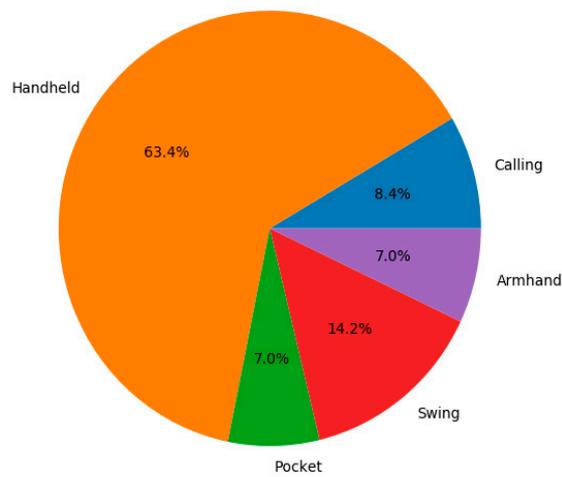


Figure 3. The distribution of the dataset.

Table 2. Performance parameters for x-IMU and smartphone.

Devices	x-IMU		Smartphone	
	Accelerometer	Gyroscope	Accelerometer	Gyroscope
Range	±16 g	±2000°/s	±8 g	±2000°/s
Stability	0.00049 g	0.06°/s	0.001 g	0.00°/s
Sample frequency	400 Hz	400 Hz	200 Hz	200 Hz

Table 3. Description of subjects.

Subjects	Gender	Age	Height (cm)	Weight (KG)
S1	M	30	169	68
S2	F	25	156	46
S3	F	25	161	53
S4	M	27	181	82
S5	M	26	173	61

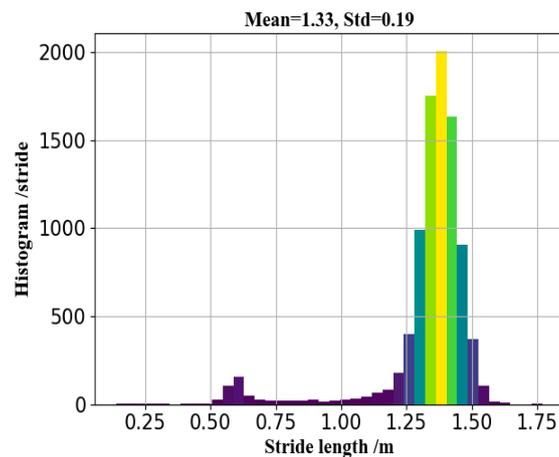


Figure 4. Histograms of real stride length.

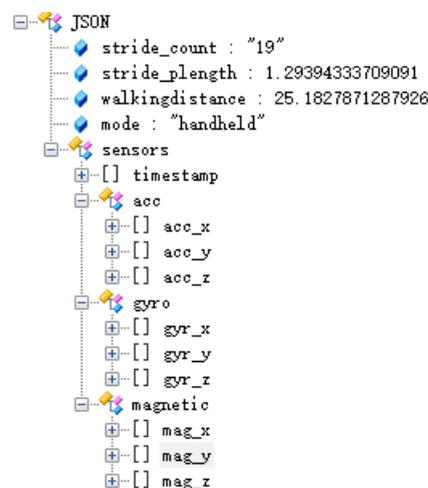


Figure 5. The JSON (JavaScript Object Notation) encapsulation of sensor data in each stride.

2.2. Pre-Processing and Walk Detection

The accelerometer data provided by the Android service were fairly noisy. High-frequency oscillations from the device and ambient environment seriously skew the clean oscillations of human motion. Normally, the step frequency was lower than 3 Hz (3 steps per second) [40]. To minimize the impact of the smartphone shaking and sensor noise, and improve the robustness of smartphone mode recognition and stride length estimation, we utilized a 1st order Butterworth filter [41] with a cutoff frequency = 3 Hz to remove the high frequency oscillations of the time-series sensors feature signal, and extract useful information from the low-cost sensor signals. Figure 6 shows the signal before and after the Butterworth filter. After using a Butterworth filter, the signal was smoother, and the insignificant parts of the signal were eliminated (see the red curve).

As shown in Figure 7, unexpectedly rotating or shaking a smartphone may arouse marked fluctuation in accelerometer and gyroscope readings, but no step event. Merely considering accelerometer and gyroscope readings for walk detection, the abnormal movements (unexpectedly rotating or shaking the smartphone) may lead to unreliable step detection results. We combined the accelerometer and gyroscope with a magnetometer to reduce the influence of random motion (shaking or rotating smartphones). This is based on the assumption that the magnitude of a magnetic reading changes significantly when the user is walking indoors, due to the magnetic field diversity at different locations. We denoted the magnitude of the gyroscope, acceleration and magnetic field at time t as g_t , a_t and m_t , respectively. We introduced a sliding window of N observed values to eliminate exception

data and consider the average magnitude of acceleration h_a , the standard deviation of the gyroscope h_g , and the standard deviation of magnetic field magnitude h_m for walk detection, as in Equations (4)–(6).

$$h_a = \frac{1}{N} \sum_{t=1}^N a_t \tag{4}$$

$$h_g = \frac{1}{N} \sum_{t=1}^N \left(g_t - \frac{1}{N} \sum_{t=1}^N g_t \right)^2 \tag{5}$$

$$h_m = \frac{1}{N} \sum_{t=1}^N \left(m_t - \frac{1}{N} \sum_{t=1}^N m_t \right)^2 \tag{6}$$

If some or all of h_a , h_g and h_m were below certain thresholds, then the user was classified as static (not walking). Otherwise, the user was identified as moving. To effectively reduce the power consumption, walking detection was used to trigger the following walking distance estimation method.

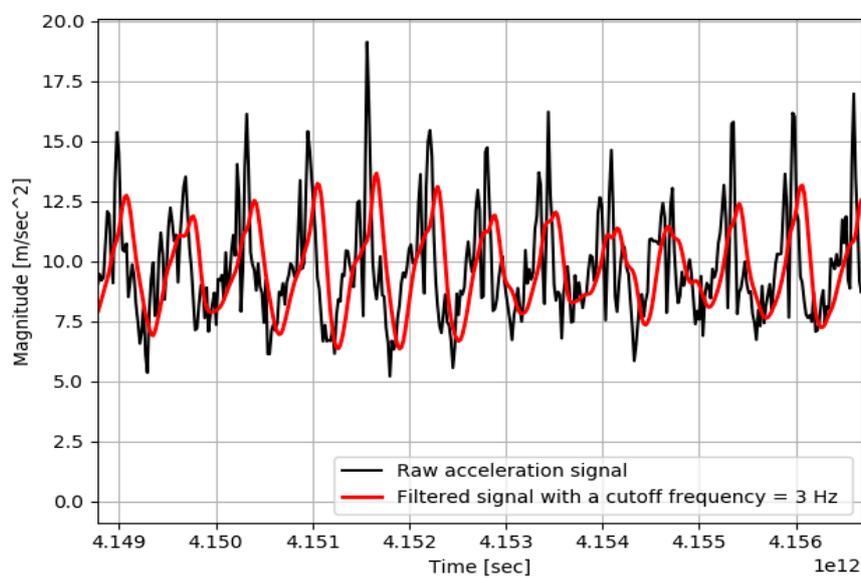


Figure 6. The signal before and after using the Butterworth filter.

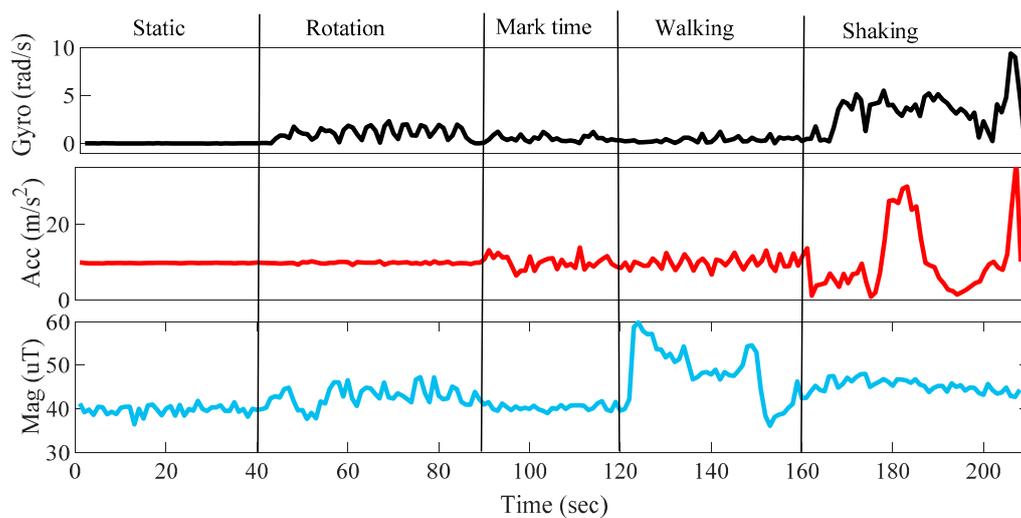


Figure 7. Walk detection with the joint decision of the gyroscope, accelerometer, and magnetometer.

2.3. Feature Extraction

Feature extraction from accelerator and gyroscope data streams is a crucial operation for smartphone mode recognition and stride length estimation. An excellent set of features provides accurate and comprehensive descriptions of motion distance. To capture either temporal variations or periodic characteristics of walking, both time-domain and frequency-domain features were considered in each stride.

- Statistical Features:** Table 4 shows the main statistical features’ description, with a brief definition of each feature, extracted from each stride observation. Mean, median, standard deviation, skewness, kurtosis, energy, maximum value, interquartile range, minimum value, and amplitude were considered.
- Time-Domain Features:** Represents how inertial sensors’ signals vary with time. Table 5 shows the time-domain features. The number of peaks, g-crossing rate, zero-crossing ratio, gyroscope-accelerometer correlation, and inter-axis correlation were extracted from each stride observation.
- Frequency-Domain Features:** Represents the inertial sensors’ signal in the frequency domain. As shown in Table 6, frequency-domain features represent signals according to their frequency components. A fast Fourier transform (FFT) was applied, and first dominant frequency, second dominant frequency, and the amplitude of the first and second dominant frequencies were the features used.
- High-Order Features:** In addition to the time-domain and frequency-domain features, we also built higher-order features based on the acknowledged studies, including Kim [24], Ladetto [22] and Weinberg [23]. All of the extracted high-order features are summarized in Table 7. The features mentioned above were extracted from the observations of accelerometer and gyroscope in one stride.

Table 4. Main statistical features’ description.

Feature	Description
Mean	The mean of a signal. $\bar{s} = \frac{\sum_{i=1}^N s_i}{N}$ where s_i are the samples, $i = 1, \dots, N$.
Standard deviation	$std = \sqrt{\frac{1}{N} \sum_{i=1}^N (\ s_i\ - \ \bar{s}\)^2}$ where s_i are the samples, $i = 1, \dots, N$.
Skewness	$s_{skew} = \frac{1}{N\sigma^3} \sqrt{\frac{1}{N} \sum_{i=1}^N (\ s_i\ - \ \bar{s}\)^3}$. Skewness is a measure of the asymmetry of the probability distribution.
Kurtosis	$s_{kurt} = \frac{1}{N\sigma^4} \sqrt{\frac{1}{N} \sum_{i=1}^N (\ s_i\ - \ \bar{s}\)^4}$. Kurtosis is a descriptor of the shape of a probability distribution.
Interquartile range	Quartiles divide an ordered data set into four equal parts. The interquartile range (IQR) is the first quartile subtracted from the third quartile.

Table 5. Main features description of the time domain.

Feature	Description
Magnitude area	The sum of absolute values of a signal.
Number of peaks	The count of maximum points within one stride window of the signal where the maximum points should be above a pre-set value.
Zero-crossing ratio	$ZCR = \frac{1}{N-1} \sum_{i=1}^{N-1} \prod \{s_i s_{i-1} < 0\}$ The zero-crossing rate is a measure of how many times within a stride a signal changes from a positive value to a negative value, and vice versa [42].
Inter-axis correlation	$R_{x,y} = \frac{N(\sum_{i=1}^N x_i y_i) - (\sum_{i=1}^N x_i)(\sum_{i=1}^N y_i)}{\sqrt{[N\sum_{i=1}^N x_i^2 - (\sum_{i=1}^N x_i)^2][N\sum_{i=1}^N y_i^2 - (\sum_{i=1}^N y_i)^2]}}$ where x_i and y_i are the samples from two axes, $i = 1, \dots, N$.
Accelerometer–gyroscope correlation	The cross-correlation coefficient between the acceleration and gyroscope.

Table 6. Main features description of the frequency domain.

Feature	Description
Spectrum energy	$energy = \sum_{i=1}^N \ s_i\ ^2$. Depicts the energy distribution of each frequency point.
Spectral entropy	Depicts the degree of uncertainty in the magnitude distribution of the source.
Frequency points	FFT (fast Fourier Transform): direct component, 1, 2, 3, 4, 5 Hz.

Table 7. Higher-order features.

Feature	Description
Weinberg	$Weinberg = \sqrt[4]{a_{max} - a_{min}}$. Weinberg utilizes the difference of the vertical acceleration values during the stride to estimate stride length. a_{max} and a_{min} represent the maximum and minimum acceleration values on the Z-axis in each stride, respectively.
Kim	$Kim = \sqrt[3]{\frac{\sum_{i=1}^N a_i }{N}}$ Kim estimate stride length based only on average acceleration during the stride. a_i represents the measured acceleration value of the i th sample in each step, N represents the number of samples corresponding to each step.
Scarlett	$Scarlett = \frac{\sum_{i=1}^N a_i }{a_{max} - a_{min}} - a_{min}$ Scarlett eliminates the spring effect of the human gait and estimates stride length based on minimum, maximum, and average acceleration.

2.4. Smartphone Mode Recognition

Once the data pre-processing and features extraction were completed, features were used to train the multi-class classifier and predict smartphone modes in a timely way.

2.4.1. Smartphone Mode Definition and Analysis

As shown in Figure 8, in addition to the normal mode of handheld, the calling, pocket, arm-hand and swinging-hand modes were also considered.

- Handheld: Pedestrian holds their phone horizontally with the hand in front of their chest while walking (see Figure 8a).
- Calling: Pedestrian makes or receives a phone call while walking (see Figure 8b).
- Trouser pocket: Pedestrian carries the smartphone in a trouser pocket while walking (see Figure 8c).
- Swinging-hand: Arm swinging is the natural motion of the arms when walking with the hands-free, and it is synchronized with the opposite side's foot (see Figure 8d).
- Arm-hand: In scenes such emergency rescue, users usually tie their smartphone to their arms (see Figure 8e).

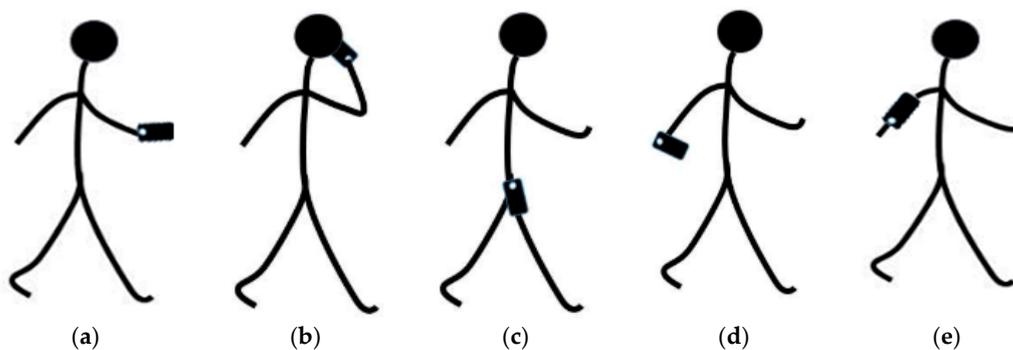


Figure 8. Different smartphone modes: (a) handheld; (b) calling; (c) trouser pocket; (d) swinging-hand; (e) arm-hand.

In consideration of the different sensor characteristics, corresponding to different smartphone modes, we analyzed the differences of inertial sensors in the five usage modes. As shown in Figures 9 and 10, the mode in the black dotted rectangle is the handheld mode; the mode in the red dotted rectangle is the calling mode; the mode in the blue dotted rectangle is the swinging-hand mode; the mode in the blue dotted rectangle is the arm-hand mode; the mode in the blue dotted rectangle is the trouser pocket mode. From the figures, we found that the observations of inertial sensors under different modes showed slight differences. Therefore, we made full use of the extracted statistical features, time-domain and frequency-domain features of inertial sensors, to identify different smartphone modes.

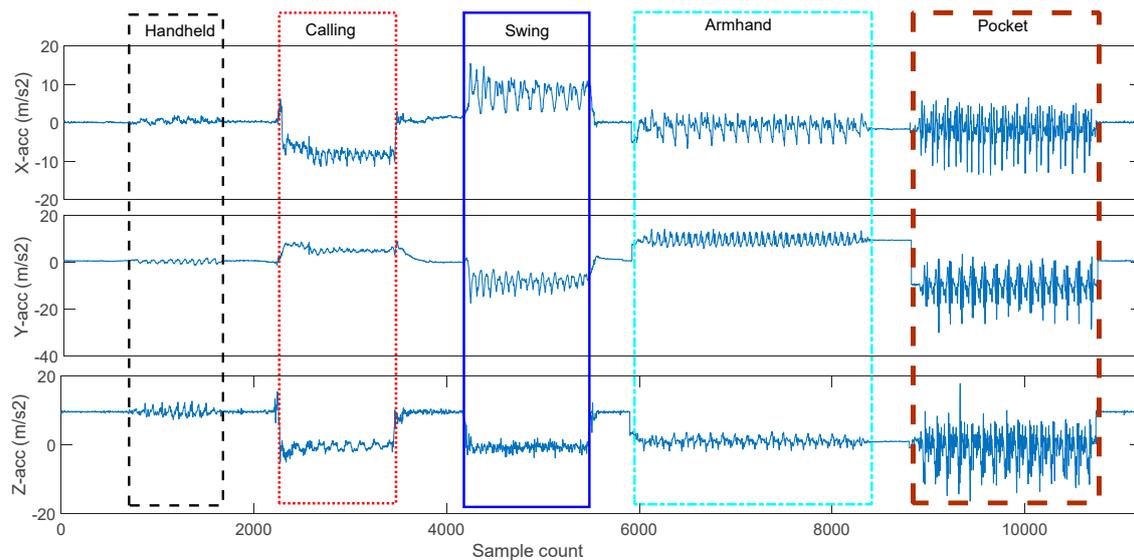


Figure 9. The triaxial acceleration of different modes.

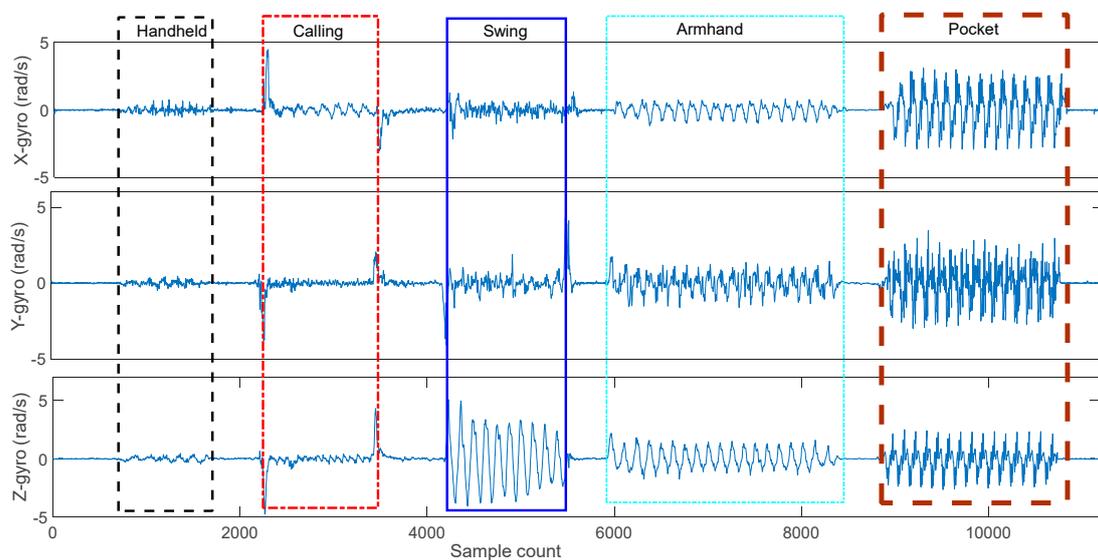


Figure 10. The triaxial gyroscope of different modes.

2.4.2. Classification Model

The key step in smartphone modes recognition is classification, which takes advantage of the extracted features. In this study, based on these features, six state-of-the-art single classifiers (Extreme Gradient Boost (XGBoost) [43], LightGBM [44], K-Nearest Neighbor (KNN) [45], Decision Tree (DT) [46],

AdaBoost [47], and support vector machines (SVM) [48]) were compared to recognize smartphone modes. Each classifier presents its advantages and disadvantages.

To improve the accuracy and robustness of smartphone mode recognition, we needed to fuse the results of multiple classifiers. Stacking is an ensemble model, where a new model is trained from the combined predictions of two (or more) previous models. In general, the stacked model outperforms each of the individual models, due to its smooth nature and ability to highlight each base model, where it performs best, and discredit each base model, where it performs poorly. As shown in Figure 11, we used a two layer stacking model for smartphone mode recognition. During the ensemble process, we utilized the predictions of non-linear models including AdaBoost [47], DT [46], KNN [45], LightGBM [44], SVM [48], and XGBoost [43] to train the first-level model to generate the second-layer train set and test set. Logistic regression in the second-level model was employed to output the final prediction.

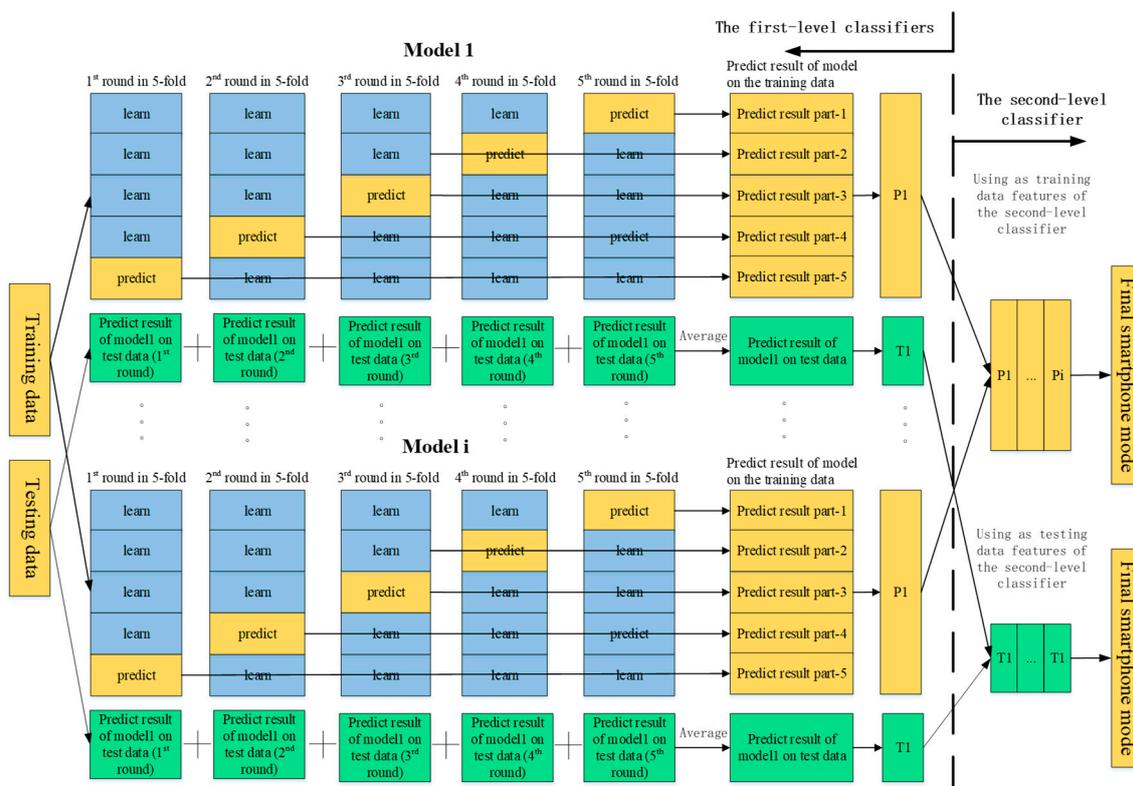


Figure 11. Stacking-based ensemble.

We took the F1 score as a performance metric to quantify the classification performance of different models. Precision is the ratio of correctly predicted conditions to the total predicted positive conditions for each class. Recall presents the ratio of correctly predicted positive conditions to all the true conditions for each class. F1 score is a combination of precision and recall that represents the detection result with less bias than the accuracy in multi-class classification problems, especially with disproportionate samples in each class [49].

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{\text{all positive predictions}}{\text{all predictions}} \tag{7}$$

$$precision = \frac{TP}{TP + FP} = \frac{\text{positive predicted correctly}}{\text{all positive predictions}} \tag{8}$$

$$recall = \frac{TP}{TP + FN} = \frac{\text{positive predicted correctly}}{\text{all positive observations}} \tag{9}$$

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (10)$$

The definitions of the above metrics use the true positive (TP), true negative (TN), false positive (FP), and false negative (FN). A high F1 score indicates a high level of classification performance and agreement between the classification and ground truth.

Figure 12 compares the accuracy of smartphone mode prediction for the six single classifiers and a stacking ensemble classifier in the three trajectories. Figure 12 indicates that the classification model based on stacking ensemble outperformed all single classifiers in the three tested trajectories. The average recognition accuracy of the classification model based on stacking ensemble reached about 98.47%. The average recognition accuracy of stacking ensemble classifier was improved by 26.3%, 1.9%, 33.5%, 22.7%, 22.8%, and 2.0% compared to AdaBoost [47], DT [46], KNN [45], LightGBM [44], SVM [48], and XGBoost [43], respectively. The precision, recall, and F-measure score of each smartphone using the stacking classifier are summarized in Table 8.

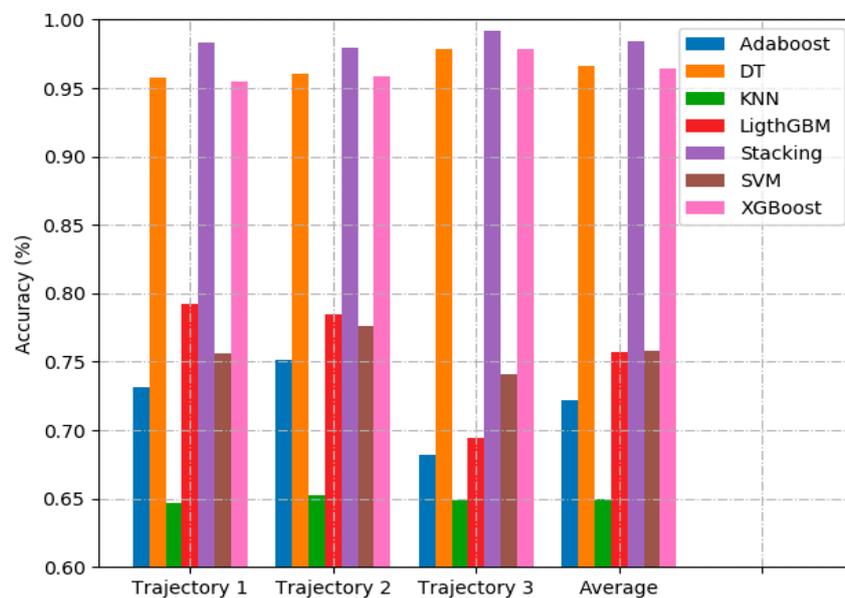


Figure 12. The comparison of recognition accuracy using stacking ensemble and single model.

Table 8. Performance evaluation of each mode with stacking classifier.

Modes	Precision (%)	Recall (%)	F-measure (%)	Support
Handheld	99.56	97.84	98.70	232
Calling	96.05	100.00	97.98	243
Pocket	97.32	99.09	98.20	110
Arm-hand	98.35	96.76	97.55	247
Swing	100.00	98.38	99.18	247
Avg/total	98.37	98.33	98.33	1079

2.5. Stride Length Estimation Based on Regression Model

2.5.1. Single Regression Models

Compared to traditional SLE methods, a regression model of machine learning has excellent generalization ability and distinct advantages in terms of approximating nonlinear continuous function. To make full use of the advantages of different machine learning models and obtain the best SLE accuracy, we trained six regression models, including Extreme Gradient Boost (XGBoost) [43], LightGBM [44], K-Nearest Neighbor (KNN) [45], Decision Tree (DT) [46], AdaBoost [47], and Support Vector Regression (SVR) [50].

2.5.2. Stacking Regression Model

To improve the accuracy and robustness of stride length estimation, the stacking regression technique of ensemble learning [51] was employed to combine multiple regression models via a meta-regressor (see Figure 13). In the offline training phase, we selected XGBRegressor [43], DecisionTreeRegressor [46], AdaBoostRegressor [47], and LightGBM [44] as single regression models. The single regression models were trained based on the complete training set. We selected SVR [50] with kernel = 'rbf' as the meta-regressor. The meta-regressor is fitted based on the outputs (meta-features) of the single regression models in the ensemble. In the online phase, the trained stacking regression model predicted the stride length of pedestrian in real time. More detail of stacking regression can be found in References [51,52].

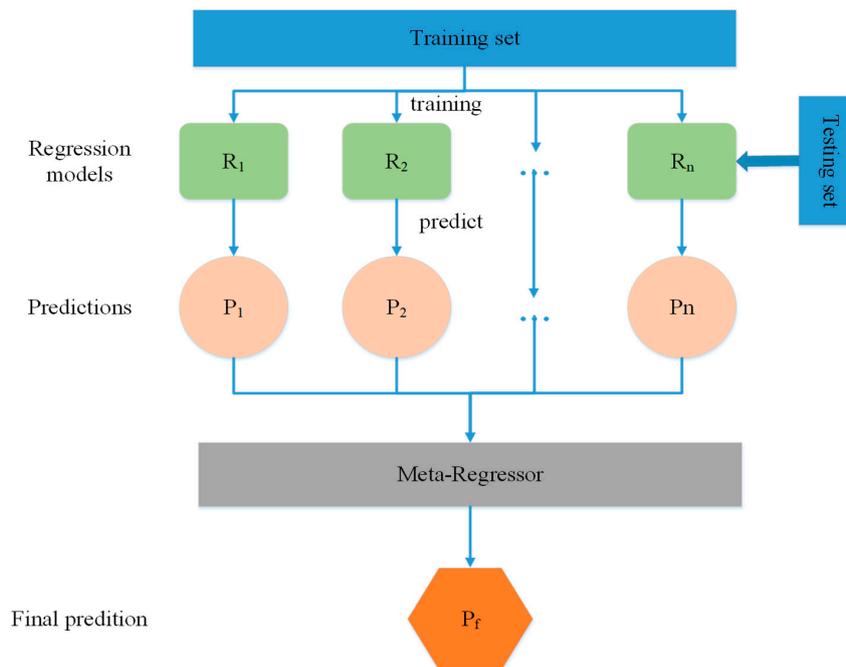


Figure 13. Stacking regressions model.

Figure 14 compares the stride length estimation of the stacking model and single regression models. The stacking model achieved the smallest SLE error in that the estimation errors of the average, the 75th percentile, and the 90th percentile were 0.039 m, 0.051 m, 0.075 m, respectively.

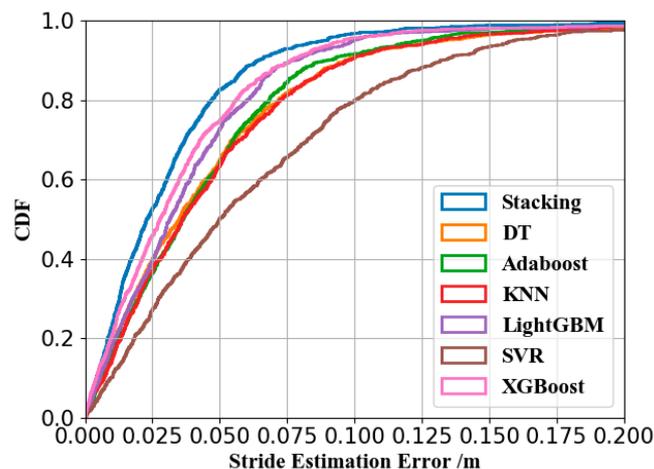


Figure 14. Comparison of stride length estimation using the stacking model and single regression models.

2.6. Walking Distance Estimation Based on Smartphone Mode Recognition

The characteristics of the inertial signals differed between the carrying modes, thus resulting in inaccurate stride length estimation. Therefore, we trained five stride length models corresponding to five smartphone modes (handheld, swing, pocket, arm-hand, and calling) in the offline phase. In the online phase, the proposed stacking classifier was used to detect smartphone mode in a timely way. Once the smartphone mode was identified, we estimated the walking distance of the pedestrian accurately by selecting the stride length model corresponding to the smartphone mode. Denoting N as the total number of strides, the walking distance D was calculated as follows:

$$D = \sum_{i=1}^N L_i \quad (11)$$

2.7. Performance Evaluation Metrics

We utilized the error rate of the stride length and walking distance as metrics to evaluate the proposed method. The error rate of the stride length was calculated with the following formula:

$$E_s = \frac{1}{N} \sum_{i=1}^N \left(\frac{|L_e^i - L_t^i|}{L_t^i} \times 100\% \right) \quad (12)$$

where L_e^i and L_t^i represent the predicted stride length and the actual stride length of the i -th stride, respectively.

The error rate of walking distance was calculated with the following formula:

$$E_{cd} = \frac{\left| \sum_{i=1}^M L_e^i - \sum_{i=1}^M L_t^i \right|}{\sum_{i=1}^M L_t^i} \times 100\% \quad (13)$$

where L_e^i and L_t^i denote the estimated stride length and the actual stride length of the i -th stride, respectively.

3. Experimentation and Evaluation

3.1. Experimental Setup

To understand the effectiveness and limitations of our proposed walking distance estimation method, we conducted a full-fledged implementation on Android to collect motion data. During the experiment, we collected data using an Android smartphone (Huawei mate 9 with 8 core 2.4 GHz processor), which was equipped with a three-axis accelerometer and a three axis gyroscope. We trained a stacking classifier and then trained five stride length models corresponding to the five smartphone modes (handheld, swing, pocket, arm-hand, and calling), respectively. We then evaluated the proposed method in both indoor and outdoor complex environments (office, stair, street, subway station, and pedestrian skyway) with natural motion patterns (fast walking, normal walking, slow walking). The smartphone mode recognition, stride length models, and walking distance estimation performance are evaluated in Sections 3.2, 3.4 and 3.5, respectively. We compared the proposed algorithm with state-of-the-art algorithms.

3.2. Experimental Results of Smartphone Mode Recognition

The five-fold cross-validation method was used to verify the performance of the proposed smartphone mode recognition method. The data set was randomly divided into five groups of the same size, where one group was retained as the validation data for testing the trained model, and the

remaining four were used as the training data. The cross-validation process was repeated five times, with each of these five groups used exactly once as the testing data.

Figure 15 demonstrates the performance of the smartphone modes recognition in the confusion matrix. The rows of the confusion matrix indicate the performed smartphone modes, while the columns indicate the predicted smartphone modes. Along the principal diagonal of the confusion matrix, the correctly classified samples are reported for each smartphone mode. Along the off-diagonal elements, the misclassified smartphone modes are reported. Figure 15 also provides the accuracy of the proposed smartphone mode recognition method. Globally, the proposed smartphone mode recognition algorithm, based on signals collected with smartphone inertial sensors, classified the modes in the correct category more than 98.10% of the time, no matter what mode was performed by the pedestrian. The average accuracy of the proposed method was 98.82%.

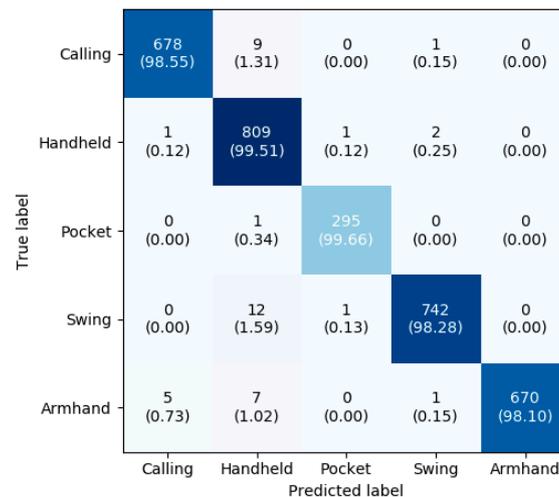


Figure 15. Recognition performance for each smartphone mode using a stacking ensemble classifier.

3.3. Comparison of Stride Length Estimation using Regression Only and Regression Based on Smartphone Mode Recognition

To explore how much performance improvement was gained from smartphone mode recognition, we trained two SLE models (regression-only-based, and regression-and-smartphone-mode-recognition-based) using the same training dataset and test data. Table 9 summarizes the stride length estimation comparison of the regression only and regression-smartphone mode recognition. A total 75% of the stride length error and error rate were 0.046 m and 3.30%, respectively. Compared to regression only, the mean error rate reduced from 5.18% to 3.30%. In other words, the mean error rate of regression-smartphone mode recognition was reduced by 36.29% ((5.18%–3.306%)/5.18%). The experimental results demonstrate that smartphone mode recognition helped improved stride length estimation accuracy.

Table 9. Comparison of stride length estimation using regression only and regression based on smartphone mode recognition.

Attributes	Regression Only		Regression Based on Smartphone Mode	
	Error	Error Rate ¹	Error	Error Rate
Mean	0.058	5.12%	0.036	3.04%
Std	0.074	-	0.048	-
25%	0.019	1.36%	0.012	0.85%
50%	0.041	3.01%	0.025	1.85%
75%	0.073	5.18%	0.046	3.30%
95%	0.144	11.31%	0.095	7.03%

¹ According to Equation (12).

3.4. Experimental Results of Stride Length Estimation

To clearly illustrate the error distribution of stride length estimation, we employed CDF (cumulative distribution function) and box plots to compare the statistics of single stride length estimation errors, as described in Figures 16 and 17. From Figure 16, we can see that the relative error of the proposed algorithm was smaller than those achieved by the Tapeline [38], Kim [24], Weinberg [23], and Ladetto [22]. In the box plots, the vertical axis and horizontal axis correspond to the SLE errors and different methods. The whiskers represent 99.3% coverage. On each box, the central (yellow) mark is the median, and the edges of the box are the 25th and 75th percentiles. From Figure 17, we can see that the median, the lower, and upper quartiles of the proposed algorithm are lower than those of the Tapeline, Kim, Weinberg, and Ladetto algorithms. Ladetto, Weinberg, and Kim have fixed model parameters that are easy to implement, but they ignore user and device heterogeneity, which leads to lower precision. By considering the pedestrian's stride frequency, the Ladetto is more accurate and robust against different walking speeds than Weinberg and Kim. Tapeline mitigates user and device heterogeneity as well as walking pattern difference by convolutional neural networks, and obtained good stride length estimation.

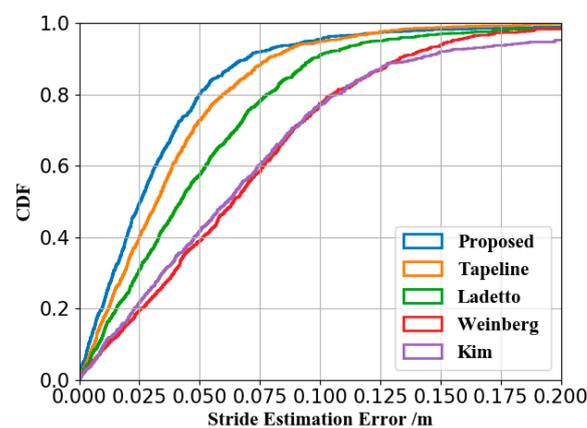


Figure 16. Comparison of stride length estimation error.

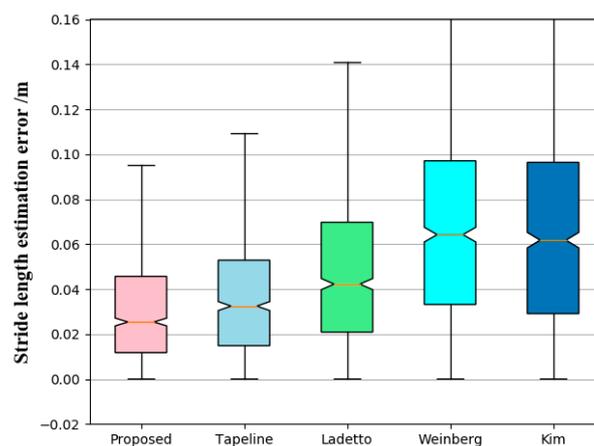


Figure 17. Box plot of stride length error.

3.5. Experimental Results of Walking Distance

The walking distance experiments were aimed to analyze the accuracy of the proposed accumulated displacement estimation method in various conditions with multiple smartphone mode changes. We started walking from an indoor office (the seventh floor of the Institute of Computing Technology, Chinese Academy of Sciences). After walking about 100 meters, we entered the stairs. We walked downstairs from the seventh floor to the ground floor, and left the office and walked along the streets

to the youth apartment of the Chinese Academy of Sciences. Figure 18 illustrates the entire trajectory. The trajectory length was 1658.73 meters and 1200 strides in total, covering office, stair, street, subway station, and pedestrian skyway. To facilitate the evaluation, we divided the entire reference trajectory into fourteen segments (see Figure 18), and annotated the scenarios and the smartphone modes (see Table 10). During each experiment, each volunteer walked along the reference trajectory precisely. To verify the adaptability of gait and smartphone placement changes, the volunteers were required to perform specific smartphone modes during walking at the specified location.

Table 10 summarizes the average accuracy of smartphone mode recognition for each segment. For instance, in segment “11,” the accuracy of smartphone mode recognition was 97.5% when the user walked through a pedestrian skyway and made a call. The comparison of cumulative distance estimation is shown in Table 11. Based on the experimental results, the error rate of the proposed walking distance estimation method was 2.62%, which is superior to Tapeline (3.28%), Ladetto (4.21%), Weinberg (6.39%), and Kim (5.25%), because our proposed method adapted to smartphone mode changes and selected the optimal stride length regression model automatically.

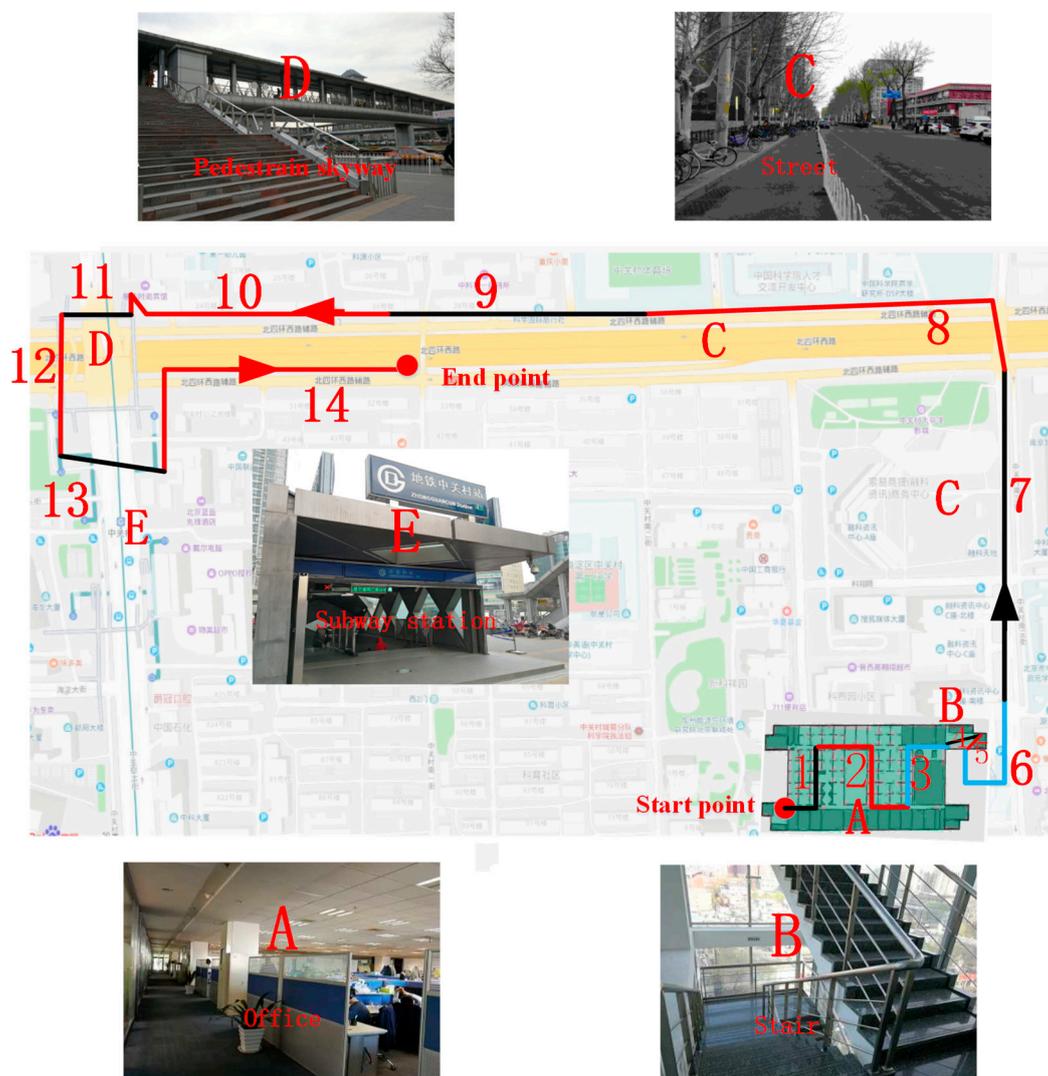


Figure 18. Walking trajectory description. The volunteer was asked to travel the highlighted trajectory with multiple smartphone mode changes.

Table 10. Accuracy of smartphone mode recognition and stride length estimation in different scenes.

Segment	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Scenarios		Office			Stair			Street			Skyway		Station	Street
Modes														
Recognition Accuracy (%)	97.2	96.6	98.4	97.3	99.1	99.6	97.8	98.6	96.8	98.9	97.5	98.7	96.9	98.5

Table 11. Comparison of cumulative distance estimation in the complex trajectory.

Attributes	Proposed	Tapeline [38]	Ladetto [22]	Weinberg [23]	Kim [24]
Avg error (m)	43.55	57.42	69.94	106.04	87.08
Avg error rate ¹ (%)	2.62	3.28	4.21	6.39	5.25

¹ According to Equation (13).

To further validate the practicality and universality of the proposed accumulated displacement estimation method, we conducted experiments in an outdoor stadium (see Figure 19) and on a road with significant inclination (see Figure 20). The stadium is a large open area. In addition, we did not set any pre-planned path. Therefore, we were free to walk without any constraints. As shown in Figure 20, the road was first about 200 meters downhill, then about 200 meters uphill. To accurately record the actual stride length of pedestrians, an FM-INS module was attached to the volunteers' insteps of the right foot. The actual accumulated displacement was calculated by summing the stride length of all strides. The actual trajectory lengths of the stadium and road were 897.76 meters (686 strides) and 366.48 meters (293 strides), respectively. The comparison of walking distance estimation is summarized in Table 12.

Table 12. Comparison of cumulative distance estimation in stadium and road with significant inclination.

Type	Attributes	Proposed	Tapeline [38]	Ladetto [22]	Weinberg [23]	Kim [24]
Outdoor stadium	error (m)	22.56	29.87	38.44	56.36	49.13
	error rate (%)	2.51	3.33	4.28	6.28	5.47
Road with inclination	error (m)	10.21	14.02	16.42	25.41	22.24
	error rate (%)	2.79	3.83	4.48	6.93	6.07

**Figure 19.** Outdoor stadium.



Figure 20. Road with significant inclination.

3.6. Complexity Analysis

We implemented the proposed method in Python with the help of Sklearn, and performed it on a personal computer equipped with an Intel Core i5-4460 CPU at 3.20 GHz and 16 GB of DDR4 RAM. The most time-consuming procedures of the proposed method were the training data collection. Training data acquisition time was equal to walking time. However, the training data collection was performed in the offline phase, meaning that they did not consume any time during the online prediction phase. Here, we only compared the time complexity with Tapeline, with similar performance. Table 13 reports the training and test time of the proposed method with Tapeline. The proposed method required only 2000 strides to train a satisfactory stride length estimation model, and the training time was 190.12 s, while Tapeline required 8000 strides, and the training time was 2 h 18 min 26s. The proposed method required only one quarter of the training samples to obtain a performance comparable to the Tapeline. Meanwhile, our proposed method was very efficient, and consumed $2278 \times$ less time in model training compared to Tapeline, because convolutional neural networks and recurrent neural networks are very time-consuming. From our test, the test times of the proposed method and Tapeline were 23 ms (23.12 s/1000) and 87 ms (86.9 s/1000) for each stride. The latency at the millisecond level was negligible.

Table 13. The time complexity analysis.

	Training Dataset Size	Test Dataset Size	Training Time	Testing Time
Smartphone mode recognition			180.99 s	20.35
Stride length regression	2000	1000	9.13 s	2.77 s
Total			3 min 10.12 s	23.12 s
Tapeline	8000	1000	2 h 18 min 26 s	86.9 s

4. Discussion and Conclusions

Concerning the inaccuracy of the traditional nonlinear method, we presented a walking distance estimation method consists of a smartphone mode recognition and stride length estimation based on the regression model using the inertial sensors of the smartphone. We proposed a smartphone mode recognition algorithm using a stacking ensemble classifier to effectively distinguish different smartphone modes, achieving an average recognition accuracy of 98.82%. The proposed walking distance estimation method obtained a superior performance, with a single stride length error rate of 3.30% and a walking distance error rate of 2.62%. The proposed method outperformed the commonly used nonlinear step length estimation method (Kim [24], Weinberg [23], Ladetto [22]) in both single

stride length and walking distance estimation. In comparison to Tapeline, this method possessed the advantages of smaller computational overhead, faster training speed, and fewer training samples. In addition to improving the performance of pedestrian dead reckoning, this technique can be used to assess the physical activity level of the user, providing feedback and motivating a more active lifestyle.

However, there are still some limitations that may be important to address in our future work. For example, only five smartphone modes (handheld, swing, pocket, arm-hand, and calling) were analyzed. More smartphone modes, such as those involving the smartphone in belts and bags, need to be further studied using similar methodologies. Additionally, we focused on the normal walk status, while other pedestrian motion states such as walking backward, lateral walking, running, and jumping will be studied in the future to construct a more viable walking distance estimation. Moreover, humans are flexible structures, it is difficult to ensure that the movement of mobile phones equals the movement of pedestrians. Extra actions (standing still, playing games, reading) result in inaccurate stride length estimation. Finally, the trained model may be not suitable for non-healthy adults (e.g., Parkinson's patients), children, and elderly. In the future, we will investigate how to obtain training data by crowdsourcing automatically, then train a personalized SLE model in the form of online learning to mitigate user and device heterogeneity.

5. Patents

A pending patent has been submitted for the proposed method.

Author Contributions: Conceptualization, Q.W., H.L. and A.M.; methodology, Q.W.; software, L.Y.; validation, H.L.; formal analysis, Q.W.; investigation, L.Y. and C.O.; resources, H.L.; data curation, L.Y.; writing—original draft preparation, Q.W.; writing—review and editing, Q.W. and C.O.; visualization, Q.W.; supervision, H.L.; project administration, F.Z. and A.M.; funding acquisition, F.Z.

Funding: This research was funded by the National Key Research and Development Program (2018YFB0505200), the BUPT Excellent Ph.D. Students Foundation (CX2018102), the National Natural Science Foundation of China (61872046, 61374214, 61671264 and 61671077) and the Open Project of the Beijing Key Laboratory of Mobile Computing and Pervasive Device.

Acknowledgments: We would like to thank the editors and the three anonymous reviewers for their valuable comments, which greatly improved the quality of this manuscript. Many thanks to the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology Chinese Academy of Sciences for the support in our research.

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

References

1. Khedr, M.; El-Sheimy, N. A Smartphone Step Counter Using IMU and Magnetometer for Navigation and Health Monitoring Applications. *Sensors* **2017**, *17*, 2573. [[CrossRef](#)] [[PubMed](#)]
2. Storti, K.L.; Pettee, K.K.; Brach, J.S.; Talkowski, J.B.; Richardson, C.R.; Kriska, A.M. Gait speed and step-count monitor accuracy in community-dwelling older adults. *Med. Sci. Sports Exerc.* **2008**, *40*, 59–64. [[CrossRef](#)] [[PubMed](#)]
3. Storm, F.A.; Heller, B.W.; Mazzà, C. Step Detection and Activity Recognition Accuracy of Seven Physical Activity Monitors. *PLoS ONE* **2015**, *10*, e0118723. [[CrossRef](#)]
4. Kuang, J.; Niu, X.; Chen, X. Robust Pedestrian Dead Reckoning Based on MEMS-IMU for Smartphones. *Sensors* **2018**, *18*, 1391. [[CrossRef](#)]
5. Wang, Q.; Luo, H.; Men, A.; Zhao, F.; Huang, Y. An Infrastructure-Free Indoor Localization Algorithm for Smartphones. *Sensors* **2018**, *18*, 3317. [[CrossRef](#)]
6. Wang, Q.; Luo, H.; Men, A.; Zhao, F.; Gao, X.; Wei, J.; Zhang, Y.; Huang, Y. Light positioning: A high-accuracy visible light indoor positioning system based on attitude identification and propagation model. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 155014771875826. [[CrossRef](#)]
7. Guo, X.; Shao, W.; Zhao, F.; Wang, Q.; Li, D.; Luo, H. WiMag: Multimode Fusion Localization System based on Magnetic/WiFi/PDR. In Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcalá de Henares, Spain, 4–7 October 2016; pp. 1–8.

8. Zhuang, Y.; Lan, H.; Li, Y.; El-Sheimy, N. PDR/INS/WiFi Integration Based on Handheld Devices for Indoor Pedestrian Navigation. *Micromachines* **2015**, *6*, 793–812. [[CrossRef](#)]
9. Li, Y.; Zhuang, Y.; Lan, H.; Zhou, Q.; Niu, X.; El-Sheimy, N. A Hybrid WiFi/Magnetic Matching/PDR Approach for Indoor Navigation with Smartphone Sensors. *IEEE Commun. Lett.* **2016**, *20*, 169–172. [[CrossRef](#)]
10. Shao, W.; Luo, H.; Zhao, F.; Crivello, A. Toward improving indoor magnetic field-based positioning system using pedestrian motion models. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1550147718803072. [[CrossRef](#)]
11. Wang, Q.; Luo, H.; Zhao, F.; Shao, W. An indoor self-localization algorithm using the calibration of the online magnetic fingerprints and indoor landmarks. In Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcalá de Henares, Spain, 4–7 October 2016; pp. 1–8.
12. Shao, W.; Zhao, F.; Wang, C.; Luo, H.; Muhammad Zahid, T.; Wang, Q.; Li, D. Location Fingerprint Extraction for Magnetic Field Magnitude Based Indoor Positioning. *J. Sens.* **2016**, *2016*, 1945695. [[CrossRef](#)]
13. Aguiar, B.; Silva, J.; Rocha, T.; Carneiro, S.; Sousa, I. Monitoring physical activity and energy expenditure with smartphones. In Proceedings of the IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), Valencia, Spain, 1–4 June 2014; pp. 664–667.
14. Diez, L.E.; Bahillo, A.; Otegui, J.; Otim, T. Step Length Estimation Methods Based on Inertial Sensors: A Review. *IEEE Sens. J.* **2018**, *18*, 6908–6926. [[CrossRef](#)]
15. Kourogi, M.; Kurata, T. A wearable augmented reality system with personal positioning based on walking locomotion analysis. In Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality, Tokyo, Japan, 10 October 2003; pp. 342–343.
16. Combettes, C.; Renaudin, V. Comparison of misalignment estimation techniques between handheld device and walking directions. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Banff, AB, Canada, 13–16 October 2015; pp. 1–8.
17. Gu, F.; Member, S.; Khoshelham, K.; Yu, C.; Shang, J. Accurate Step Length Estimation for Pedestrian Dead Reckoning Localization Using Stacked Autoencoders. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 1–9. [[CrossRef](#)]
18. Jahn, J.; Batzer, U.; Seitz, J.; Patino-Studencka, L.; Gutierrez Boronat, J. Comparison and evaluation of acceleration-based step length estimators for handheld devices. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation, Zurich, Switzerland, 15–17 September 2010; pp. 1–6.
19. Ho, N.H.; Truong, P.; Jeong, G.M. Step-Detection and Adaptive Step-Length Estimation for Pedestrian Dead-Reckoning at Various Walking Speeds Using a Smartphone. *Sensors* **2016**, *16*, 1423. [[CrossRef](#)] [[PubMed](#)]
20. Pepa, L.; Marangoni, G.; Di Nicola, M.; Ciabattini, L.; Verdini, F.; Spalazzi, L.; Longhi, S. Real time step length estimation on smartphone. In Proceedings of the IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 7–11 January 2016; pp. 315–316.
21. Jiang, Y.; Li, Z.; Wang, J. PTrack: Enhancing the Applicability of Pedestrian Tracking with Wearables. *IEEE Trans. Mob. Comput.* **2019**, *18*, 431–443. [[CrossRef](#)]
22. Ladetto, Q.; Terrier, B.M.; Philippe, T.; Schutz, Y. On foot navigation: Continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering. *J. Navig.* **2000**, *53*, 279–285. [[CrossRef](#)]
23. Weinberg, H. Using the ADXL202 in Pedometer and Personal Navigation Applications. Available online: <http://www.bdtic.com/Download/ADI/AN-602.pdf> (accessed on 17 February 2019).
24. 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. [[CrossRef](#)]
25. Allseits, E.; Agrawal, V.; Lučarević, J.; Gailey, R.; Gaunaud, I.; Bennett, C. A practical step length algorithm using lower limb angular velocities. *J. Biomech.* **2018**, *66*, 137–144. [[CrossRef](#)]
26. Shin, S.H.; Park, C.G. Adaptive step length estimation algorithm using optimal parameters and movement status awareness. *Med. Eng. Phys.* **2011**, *33*, 1064–1071. [[CrossRef](#)]
27. Hannink, J.; Kautz, T.; Pasluosta, C.F.; Barth, J.; Schulein, S.; GaBmann, K.G.; Klucken, J.; Eskofier, B.M. Mobile Stride Length Estimation with Deep Convolutional Neural Networks. *IEEE J. Biomed. Health Inform.* **2018**, *22*, 354–362. [[CrossRef](#)] [[PubMed](#)]
28. Cho, S.Y.; Park, C.G. MEMS Based Pedestrian Navigation System. *J. Navig.* **2006**, *59*, 135–153. [[CrossRef](#)]
29. Xing, H.; Li, J.; Hou, B.; Zhang, Y.; Guo, M. Pedestrian Stride Length Estimation from IMU Measurements and ANN Based Algorithm. *J. Sens.* **2017**, *2017*, 6091261. [[CrossRef](#)]

30. Liu, Y.; Chen, Y.; Shi, L.; Tian, Z.; Zhou, M.; Li, L. Accelerometer Based Joint Step Detection and Adaptive Step Length Estimation Algorithm Using Handheld Devices. *J. Commun.* **2015**, *10*, 520–525. [[CrossRef](#)]
31. Kang, J.; Lee, J.; Eom, D.S. Smartphone-Based Traveled Distance Estimation Using Individual Walking Patterns for Indoor Localization. *Sensors* **2018**, *18*, 3149. [[CrossRef](#)] [[PubMed](#)]
32. Zhu, Z.; Wang, S. A Novel Step Length Estimator Based on Foot-Mounted MEMS Sensors. *Sensors* **2018**, *18*, 4447. [[CrossRef](#)]
33. Martinelli, A.; Gao, H.; Groves, P.D.; Morosi, S. Probabilistic Context-Aware Step Length Estimation for Pedestrian Dead Reckoning. *IEEE Sens. J.* **2018**, *18*, 1600–1611. [[CrossRef](#)]
34. Munoz Diaz, E. Inertial Pocket Navigation System: Unaided 3D Positioning. *Sensors* **2015**, *15*, 9156–9178. [[CrossRef](#)]
35. Diaz, E.M.; Gonzalez, A.L.M. Step detector and step length estimator for an inertial pocket navigation system. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, South Korea, 27–30 October 2014; pp. 105–110.
36. Zihajehzadeh, S.; Park, E.J. Experimental evaluation of regression model-based walking speed estimation using lower body-mounted IMU. In Proceedings of the 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Orlando, FL, USA, 16–20 October 2016; pp. 243–246.
37. Zihajehzadeh, S.; Park, E.J. Regression Model-Based Walking Speed Estimation Using Wrist-Worn Inertial Sensor. *PLoS ONE* **2016**, *11*, e0165211. [[CrossRef](#)]
38. Wang, Q.; Ye, L.; Luo, H.; Men, A.; Zhao, F.; Huang, Y. Pedestrian Stride-Length Estimation Based on LSTM and Denoising Autoencoders. *Sensors* **2019**, *19*, 840. [[CrossRef](#)]
39. x-IMU Sensor Board. Available online: <http://x-io.co.uk/x-imu/> (accessed on 15 April 2019).
40. Cavagna, G.A.; Franzetti, P. The determinants of the step frequency in walking in humans. *J. Physiol.* **1986**, *373*, 235–242. [[CrossRef](#)]
41. Butterworth, S. On the theory of filter amplifiers. *Wirel. Eng.* **1930**, *7*, 536–541.
42. Saeedi, S.; Moussa, A.; El-Sheimy, N. Context-Aware Personal Navigation Using Embedded Sensor Fusion in Smartphones. *Sensors* **2014**, *14*, 5742–5767. [[CrossRef](#)] [[PubMed](#)]
43. Chen, T.; Guestrin, C. XGBoost: Reliable Large-scale Tree Boosting System. *arXiv* **2016**, arXiv:1603.02754.
44. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Long Beach, CA, USA, 2017; pp. 3146–3154.
45. Cover, T.M.; Hart, P.E. Nearest Neighbor Pattern Classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [[CrossRef](#)]
46. Quinlan, J.R. Induction of Decision Trees. *Mach. Learn.* **1986**, *1*, 81–106. [[CrossRef](#)]
47. Freund, Y.; Schapire, R.R.E. Experiments with a New Boosting Algorithm. *Int. Conf. Mach. Learn.* **1996**, *96*, 148–156.
48. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
49. Banos, O.; Galvez, J.M.; Damas, M.; Pomares, H.; Rojas, I. Window Size Impact in Human Activity Recognition. *Sensors* **2014**, *14*, 6474–6499. [[CrossRef](#)]
50. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [[CrossRef](#)]
51. Breiman, L. Stacked regressions. *Mach. Learn.* **1996**, *24*, 49–64. [[CrossRef](#)]
52. Raschka, S. Mlxtend 0.9. Available online: <https://sebastianraschka.com/pdf/software/mlxtend-latest.pdf> (accessed on 29 April 2019).

