

RESEARCH ARTICLE

# Prediction of abnormal gait behavior of lower limbs based on depth vision

Tie Liu<sup>1</sup> , Dianchun Bai<sup>1</sup>, Hongyu Yi<sup>2</sup> and Hiroshi Yokoi<sup>3</sup>

<sup>1</sup>School of Electrical Engineering, Shenyang University of Technology, Shenyang 110870, China

<sup>2</sup>Shenyang Fire Research Institute of Emergency Management Ministry, Shenyang 110034, China

<sup>3</sup>Department of Mechanical Engineering and Intelligent Systems, The University of Electro-Communications, Chofu 182-8585, Japan

**Corresponding author:** Dianchun Bai; Email: [baidianchun@sut.edu.cn](mailto:baidianchun@sut.edu.cn)

**Received:** 29 January 2024; **Revised:** 26 April 2024; **Accepted:** 11 May 2024

**Keywords:** electronic walker; machine vision; depth learning; behavior prediction

## Abstract

As a kind of lower-limb motor assistance device, the intelligent walking aid robot plays an essential role in helping people with lower-limb diseases to carry out rehabilitation walking training. In order to enhance the safety performance of the lower-limb walking aid robot, this study proposes a deep vision-based abnormal lower-limb gait prediction model construction method for the problem of abnormal gait prediction of patients' lower limbs. The point cloud depth vision technique is used to acquire lower-limb motion data, and a multi-posture angular prediction model is trained using long and short-term memory networks to build a model of the user's lower-limb posture characteristics during normal walking as well as a real-time lower-limb motion prediction model. The experimental results indicate that the proposed lower-limb abnormal behavior prediction model is able to achieve a 97.4% prediction rate of abnormal lower-limb movements within 150 ms. Additionally, the model demonstrates strong generalization ability in practical applications. This paper proposes further ideas to enhance the safety performance of lower-limb rehabilitation robot use for patients with lower-limb disabilities.

## 1. Introduction

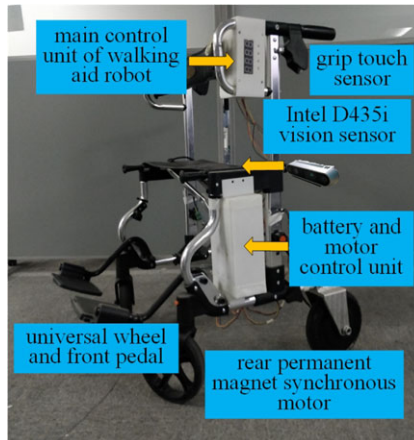
In recent years, the increasing aging population, rising incidence of stroke, and a surge in accidental injuries, particularly those related to traffic accidents, have led to a growing number of patients with physical mobility impairments [1, 2], and a growing population is experiencing varying degrees of loss of mobility, relying on others for long-term care. Therefore, lower-limb rehabilitation training has become increasingly important for these patients [3, 4]. Exoskeleton robots can assist patients with lower-limb diseases in providing functions such as standing and walking [5, 6]. Oyman EL et al. designed a cable-driven rehabilitation robot that can be easily configured to exercise different joints but is limited by space constraints [7]. Ye et al. developed a multi-posture lower-limb rehabilitation robot system that utilizes an online personalized classifier for real-time action recognition and corresponding rehabilitation training. However, the action classification process takes slightly longer [8]. Wang Y et al. studied the mechanical characteristics of cable-driven lower-limb rehabilitation robots (CDLR) and established the dynamic model of CDLR, but further research is needed on the safety assessment of this system and compliance control strategies [9]. Wang JH proposed the development of an end-effector-type full-body walking rehabilitation robot and three walking training methods. However, this study is greatly influenced by the environment and is not suitable for complex surroundings [10]. Gonçalves RS developed three non-maneuverable individual lower-limb joint rehabilitation mechanisms based on a four-bar linkage mechanism and built a prototype of the knee joint mechanism for initial experimental testing. However, this study did not involve real-assistive walking for patients [11]. Hwang used a cane-walking

exoskeleton robot, and the wearer walked with environmental recognition between the cane support point and the foot position of the supporting leg. However, this study only involved one subject and lacks good generalizability [12]. Chawin et al. introduced an active assistive trunk support system, indicating its potential suitability for individuals who cannot independently sit due to trunk impairments, thereby promoting active trunk movement assistance [13]. Li et al. designed a novel soft-rigid knee joint-assistive robotic system (SR-KR) and conducted mechanical performance testing, demonstrating its potential in the field of rehabilitation [14]. The lower-limb walking-assist robot used in this study provides patients with walking support, walking assistance, and impedance training functions.

As the popularity of rehabilitation robots in gait rehabilitation increases [15], gait cycle disturbance analysis based on different muscle analyses becomes more critical [16]. Zhang P proposed a fuzzy radial impedance (RBF-FVI) controller and developed a six-degree-of-freedom lower-limb rehabilitation exoskeleton. However, this exoskeleton system did not detect human falls [17]. Yuqi W designed a cable-driven waist rehabilitation training parallel robot, but this exoskeleton system did not detect falls during assisted walking [18]. Ercolano proposed a method for recognizing daily activities based on deep learning and skeleton data. However, this method did not run in a real environment [19]. Qin proposed a least squares support vector regression (LS-SVR) prediction algorithm to predict the gait data of the human lower limbs at the next moment. However, this method did not collect a substantial amount of lower-limb posture data [20]. S. K. Challa proposed a human gait trajectory generator based on Long Short-Term Memory (LSTM) to capture human gait data during treadmill walking. However, this model did not thoroughly study the effectiveness of trajectory generation [21]. Yu et al. proposed a gait phase detection system based on inertial measurement units (IMUs). Experimental results showed that the proposed system can identify the gait phases of stroke survivors. However, the system does not include a fall detection functionality [22]. In the current stage, walkers primarily assist users in walking, without monitoring the user's gait or preventing the user from falling.

Due to gait being considered a well-known visual recognition technique [23], three-dimensional vision cameras are more robust and less affected by lighting conditions compared to traditional two-dimensional camera recognition. V. B. Semwal employed the muscle-skeletal model in OpenSim to calculate the real-time inverse kinematics of the leg's three-link motion, proposing a learning-based approach using LSTM models for gait generation. This method aims to overcome the limitations of model-based approaches but has not been applied in practical walking aids [24]. Salvatore Gaglio and colleagues presented a method using RGB-D cameras to perceive information for identifying human activities, estimating multiple relevant joints within the human body. Experimental results demonstrate that this estimation method can detect the postures involved in various activities [25]. Walking assistive robots equipped with depth cameras continuously monitor the lower-limb posture and user's movement status in real time to determine the risk of abnormal lower-limb behavior. This enables prompt intervention of lower-limb assistive robots to provide protection for users [26]. Wei Xu and collaborators improved the accuracy of two-dimensional and three-dimensional posture estimation in the CAREN system using multi-angle videos, showing high precision in estimating human body posture for clinical applications [27]. The walking aid robot used in this study, combined with a 3D depth vision camera, can instantly detect the user's movement posture when using the walking aid, ensuring user safety.

Recent researchers have explored the use of depth vision sensors to monitor human abnormal behaviors. Jack White and others proposed a novel end-to-end visual processing pipeline for artificial vision. This pipeline employs deep reinforcement learning to learn important visual filters in offline simulations, which are later deployed on artificial vision devices capable of processing images captured by cameras and generating real-time task-guided scene representations [28]. Lei Yang and colleagues utilized depth values to separate background information, analogizing the human body as an elliptical model. They determine whether the human body has undergone abnormal movement based on the angle between the ground and the elliptical model [29]. Choon Kiat and team established a dataset of human lower-limb behavior using the Kinect skeleton tracking model. They used the vertical velocity threshold of the centroid of skeletal points as a criterion for monitoring the trend of falling [30]. In real-world scenarios, subjects are often obstructed by different static and dynamic objects, leading to the loss of gait data. This



**Figure 1.** Walking aid robot platform.

obstruction is referred to as occlusion [31]. Three-dimensional vision cameras provide a simple method for collecting multidimensional motion signals.

To enhance the safety performance of individuals with lower-limb impairments using lower-limb assistive robots, this study innovatively combines depth vision and deep learning to construct a model for predicting user lower-limb behavior. Given that depth cameras can capture rich lower-limb pose information within short distances and small areas, they are utilized to monitor local leg behaviors of the user. Simultaneously, models trained using deep learning are employed to enhance the robot's ability to detect user lower-limb abnormalities within a relatively short period, ultimately achieving the prediction of user lower-limb abnormal actions and thus improving the safety performance of the lower-limb assistive device.

Section II of this thesis contains the data collection and preprocessing of the data and illustrates the network structure construction. Section III contains the experimental results and discussion of the results. Section IV contains the discussion of this thesis. Section V contains the conclusion and future work.

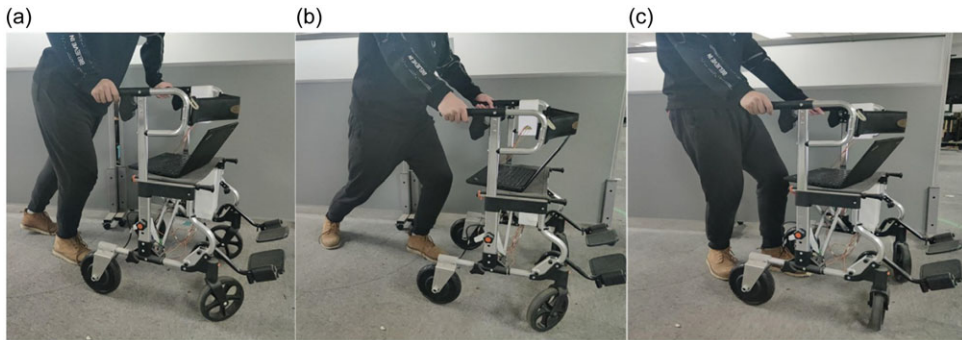
## 2. Materials and methods

The experimental equipment of this study is a self-developed walking aid, which can collect the attitude angle data of lower limbs in real time. The physical object is shown in Fig. 1.

The Intel D435i camera was selected to acquire lower-limb point cloud data with an operating range (min ~ max) of  $-3\text{ m} \sim 3\text{ m}$ , depth resolution of FPS 1280 x 720 30 fps, field of view depth of H:87 V:58, with an RGB sensor, module dimensions of 90m x 25m x 25m (Camera), and system interface type of USB 3.

The walking aid robot has three working modes: forward assistance mode, active resistance training mode, and slow electronic braking mode.

When the user is using the lightweight walking robot normally, both hands will be on the handle to control the usage mode of the walking robot. When the user has an emergency or falls, both hands or one hand will disengage from the grip and the robot will stop immediately. This study predicts the tendency to fall without disengaging the handle and does not consider the above situation. The abnormal gait movements of the user before the fall can be classified into three categories: forward leaning abnormal, unilateral abnormal, and backward leaning abnormal. The three abnormal states are shown in Fig. 2.



**Figure 2.** Common abnormal gait before falling: (a) Forward tilt abnormality; (b) Unilateral abnormality; (c) Backward tilt abnormality.

When users engage in any of the aforementioned risky behaviors while using the lower-limb assistive walker, if one or both hands detach from the handles, the two motors of the walker's rear wheels will promptly and safely come to a halt, providing support to the user and ensuring their safety. Once the user's risk is mitigated and they are capable of independently participating in rehabilitation training, simply placing both hands back onto the handles will allow the lower-limb rehabilitation walker to resume monitoring the user's lower-limb behavior to anticipate any risky movements.

### **2.1. Collect point cloud data of human lower-limb posture**

The Intel D435i camera is selected to collect the point cloud data of lower limbs, the point cloud data source was 15 health subjects including 6 females and 9 males, subjects are informed and voluntary, aged between 23–26 years, with an average age of 24.3 ( $\pm 1.03$ ) years, height of 172.1 ( $\pm 6.46$ ) cm, and weight of 71.3 ( $\pm 9.58$  kg.). The subjects were not trained prior to the test. The number of depth camera acquisitions, duration, average ankle and knee joint amplitudes are shown in Table 1.

The experimental data of these 15 individuals are divided into two parts: data from 5 individuals are used to train the LSTM neural network, while data from the other 10 individuals are utilized to validate the real-time performance and accuracy of the trained model. Place the Intel D435i depth sensor on the central axis of the frame and place it facing the user. Next, adjust the camera placement height to collect the point cloud data of the whole leg, and confirm that there is no occlusion in the field of view. The experimental software environment of walking motion information acquisition algorithm based on three-dimensional vision is Windows10, Visual Studio 2019, PCL 1.11.0. The hardware environment is Intel Core TM i7-9850h CPU@2.59 GHz, NVIDIA RTX 4000 graphics processor.

After the point cloud data are collected by the three-dimensional vision camera, the three-dimensional coordinate system is obtained through transformation so as to describe the spatial position of the measured object. The data acquisition of lower-limb depth is shown in Fig. 3.

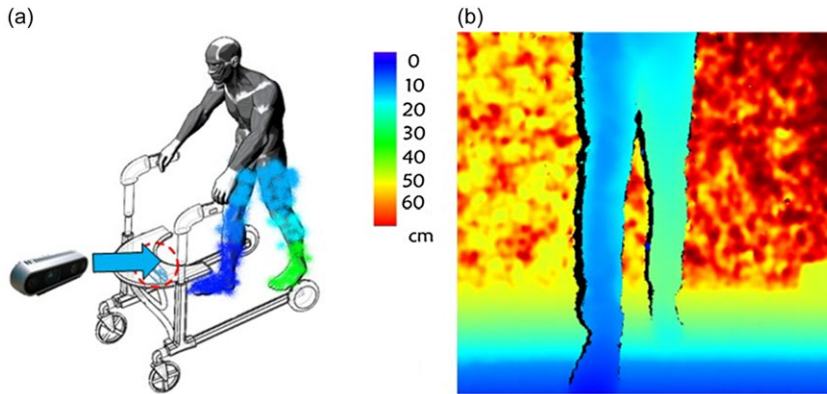
Since the depth vision has a great deal of redundant information after collecting the raw information of the lower limb, it is impossible to extract the significant features of the lower limb directly. So this raw information needs to be processed to extract features.

The collected panoramic 3D data are shown in Fig. 4. The acquisition of panoramic point cloud data map is subject to coordinate conversion, and the color frame image is aligned with the depth frame image to represent the depth coordinates in pixel form.

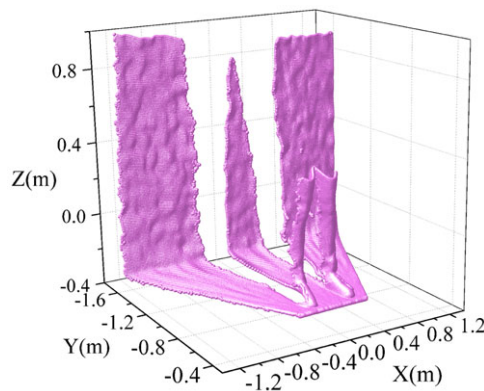
Let  $p_{\text{pixel}}(u, v, 1)$  be a specific point in the pixel coordinate system, convert the pixel coordinate information into spatial position information, and the transformed coordinate system can be called spatial coordinate system. Among them, the point coordinate corresponding to the pixel coordinate system

**Table I.** Sensors collect relevant parameters.

	Repetitions	Duration(s)	Hip joint amplitude (cm)	Knee joint amplitude (cm)
Forward tilt abnormality	10	2.7	72.5	41.8
Unilateral abnormality	10	3.2	69.3	38.3
Backward tilt abnormality	10	3.6	65.7	36.4



**Figure 3.** Diagram of depth data acquisition method: (a) Lower-limb behavior data acquisition; (b) Depth camera acquisition diagram.



**Figure 4.** Global point cloud data.

$p_{\text{pixel}}$  is  $p_{\text{img}}(x_i, y_i, 1)$ ; then

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/dx & 0 & 0 \\ 0 & 1/dy & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{1}$$

In Eq. (1),  $dx$  is the unit length of the pixel and  $dy$  is the unit width of the  $p_{\text{pixel}}$ , so that:

$$X = \begin{bmatrix} 1/dx & 0 & 0 \\ 0 & 1/dy & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The relationship between  $p_{\text{pixel}}$  and  $p_{\text{img}}$  is obtained as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{X} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (3)$$

The second transformation coordinate system is the transformation from the picture coordinate system to the point cloud coordinate system. Set a point  $P_i(x_i, y_i, z_i)$  as the coordinate of point  $P$  in the point cloud coordinate system. The relationship between  $P_{\text{img}}$  and  $P_i$  is:

$$z_i P_{\text{img}} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} P_i \quad (4)$$

Order:

$$\mathbf{H} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

In the  $H$  matrix, the elements  $f_x, f_y, c_x, c_y,$  and  $s$  are camera internal parameters, where  $f_x$  and  $f_y$  are focal lengths, which are generally equal;  $c_x$  and  $c_y$  are primary coordinates relative to the imaging plane;  $s$  is the coordinate axis inclination parameter, which is ideally 0. The elements  $f_x, f_y, c_x, c_y,$  and  $s$  are usually set by the manufacturer when the camera leaves the factory. After conversion, the coordinates of  $P_i$  point in the point cloud coordinate system are:

$$P_i = z_i \mathbf{H}^{-1} P_{\text{img}} = z_i \mathbf{H}^{-1} \mathbf{X}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (6)$$

Order:

$$\mathbf{K}^{-1} = z_i \mathbf{H}^{-1} \mathbf{X}^{-1} \quad (7)$$

Given that the depth value of  $p_i$  point is  $z_i$ , the conversion relationship between point cloud coordinate system and pixel coordinate system is as follows:

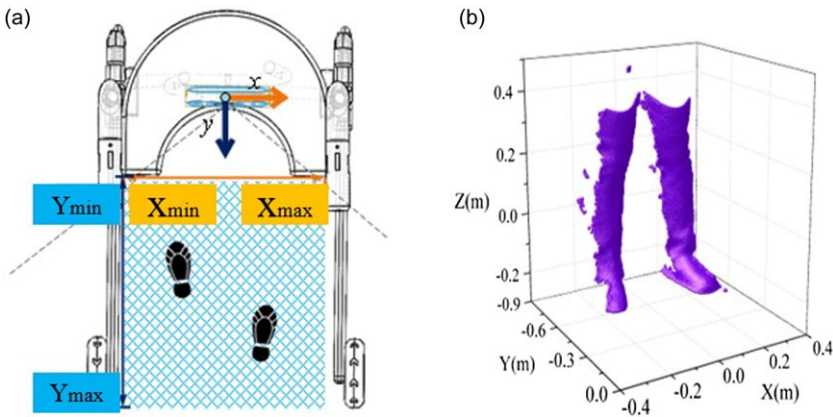
$$P_i = \mathbf{K}^{-1} P_{\text{pixel}} \quad (8)$$

## 2.2. Optimization of lower-limb attitude point cloud data

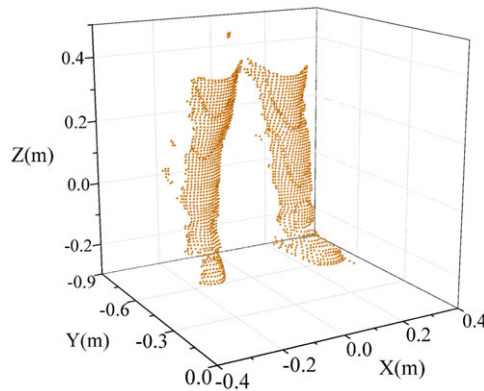
The whole point cloud data contain available information of lower-limb point cloud data, background point cloud, noise point cloud, and other redundant data. These redundant data will occupy a large amount of computational resources and storage space, which need to be optimized for the whole point cloud data so as to improve the real-time performance of the effective data acquisition of the lower-limb point cloud. Therefore, this study will not directly use point cloud data as the original data for neural network model training. In this study, a three-layer point cloud data filter is designed, containing straight-pass filtering, body pixel sampling, and statistical filtering.

### 2.2.1. Point cloud direct filtering

When the user and the walking aid robot move forward at the same time, take the point cloud coordinate system as the origin to establish the lower-limb point cloud space state diagram, as shown in Fig. 5(a).



**Figure 5.** Schematic diagram of pass-through filter: (a) Boundary condition; (b) Through filter point cloud.



**Figure 6.** Point cloud data after voxel grid filter.

After direct filtering, the point cloud is shown in Fig. 5(b), and the contour information of leg point cloud data is obtained.

The point cloud segmentation boundary conditions can be shown by Eq. (9), where  $x_{min}$  and  $x_{max}$ , respectively, represent the minimum and maximum values of the segmentation boundary  $x$ ,  $y_{min}$ , and  $y_{max}$ , respectively, represent the minimum and maximum values of the segmentation boundary  $y$  so as to avoid blocking the original point cloud data.

$$\begin{cases} -25cm < x_i < 25cm \\ -27.5cm < y_i < 27.5cm \end{cases} \quad (9)$$

In Eq. 9,  $x_i$  and  $y_i$  represent the horizontal and vertical coordinates of the user's position within the point cloud segmentation boundary, respectively.

### 2.2.2. Voxel downsampling

The amount of data after straight-through filtering is still relatively large, which leads to a decrease in the real-time performance of lower-limb pose resolution. To solve this problem, a body voxel downsampling algorithm is used, which is less complex, saves some storage space, and preserves the detailed information of the point cloud data. The sampled point cloud data are shown in Fig. 6.

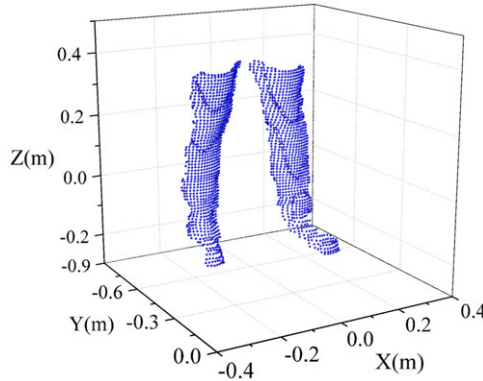


Figure 7. Point cloud data after statistical filter.

Let the global enclosing space containing all point cloud data be  $S$ , and the enclosing space be  $m$  cubes of width  $w$ . These cubes will fill the whole point cloud enclosing space as shown in Eq. 10.

$$S = m \times w^3 \tag{10}$$

The point cloud data corresponded to the cubes at the corresponding positions, and the invalid cubes that do not contain point clouds are removed. In the valid cube, there is one or more point cloud data, and all the point cloud data in the original cube are replaced by finding the center of gravity value of the point cloud data in each valid cube, so that the body pixel sampling is completed. When the value of  $w$  is larger, the value of  $m$  is smaller, which improves the operation speed, but the details of the original data will be lost; when the value of  $w$  is smaller, the value of  $m$  is larger, which is able to contain a large amount of point cloud data information, but will reduce the operation speed. In practice, the  $w$  value will be adjusted according to the computing power of the processor, and the parameter  $w$  set needs to be traded off in both sampling fidelity and real-time. This walker could be moved in the x-axis direction with a range of 50 cm, y-axis with a range of 55 cm, and z-axis with a range of 70 cm.  $w$  was selected to be the best when 0.2 cm was chosen after many tests, and the  $m$ -value at this time was 24062500.

2.2.3. Statistical filtering

After direct filtering and volume voxel downsampling, there are still some separation points or noise points in the point cloud data. In order to remove the separation points or noise points, the neighborhood point cloud filter is used. Build a KD tree (k-dimensional tree) according to the effective point cloud data  $p_k(k = 1, 2, \dots, N)$ , calculate the k-nearest neighbor  $d_k(k = 1, 2, \dots, N)$  for each dataset, calculate the distance of the k-nearest neighbor, and calculate the mean  $\bar{d}$  and variance  $\sigma^2$  of all  $d_k$

$$\bar{d} = \frac{1}{N} \sum_{k=1}^N d_k \tag{11}$$

$$\sigma^2 = \frac{1}{N} \sum_{k=1}^N (d_k - \bar{d})^2 \tag{12}$$

Due to the distance between the departure point and the effective point cloud cluster, the boundary condition  $d_{efc}$  can be established to make  $d_{efc} = \bar{d} + \alpha\sigma$ . For each point, if its K-nearest neighbor average distance  $d_k > d_{efc}$ , it is regarded as a separation point and deleted. The point cloud data after statistical filtering are shown in Fig. 7.

As shown in Fig. 7, complete, accurate, real-time, and fast lower-limb posture data could be obtained after statistical filtering of lower-limb posture point cloud data.



A variety of abnormal real-time point cloud data of human lower limbs are obtained after direct filtering, volume voxel downsampling, and statistical filtering. The abnormal posture data of human lower limbs with small amount of data and clear characteristics are used to establish the prediction dataset of abnormal posture angle of lower limbs.

### 2.3. Constructing a model to monitor trends in abnormal lower-limb gait

When the patient walks normally with the walking aid robot, the prediction model trained by using deep learning has higher accuracy and is suitable for the calculation of the walking aid robot. When the user has a tendency to have abnormal lower-limb movements, the data input to the model at that time will lead to abnormal computation. The time domain value of the lower-limb posture angle of the real-time prediction model can be used as a feature to determine whether the lower limb shows abnormal behavior and to improve the rapidity of abnormal gait trend monitoring.

Using DTW algorithm to plan the element relationship of different time series, it is possible to solve the problems of local expansion and time axis drift of multidimensional time series [32]. Set the time axis window data of the bending angle of the knee joint of the right leg during walking as  $\alpha = [x_1, x_2, x_3 \dots, x_m]$  and the predicted time axis window data as  $\beta = [y_1, y_2, y_3 \dots, y_n]$ . Define the dimension of local distance matrix  $m$  as  $m \times n$ ;  $(x_i, y_i)$  is any point in the matrix. Each element in the matrix is shown in Eq. (13).

$$d_{i,j} = \|x_i - y_j\|, i \in [1, m], j \in [1, n] \tag{13}$$

The planned path is  $P$ , indicating  $\alpha$  and  $\beta$  element  $p_k$  in vector  $p = (m, n) k$ , where:

$$P = [p_1, p_2, \dots, p_k], \max(m, n) \leq k \leq m + n - 1 \tag{14}$$

Path  $P$  must meet three constraints:

(1) Boundary condition: the planned path is in matrix  $M$ , starting from point  $p_1 = (1,1)$  and ending at point  $p_k = (m, n)$  of matrix  $M$ .

(2) Monotonicity condition: in order to ensure the monotonicity of the planned path along the time axis, the two adjacent elements in path  $p, p_k = (a, b)$  and  $p_{k+1} = (a', b')$  need to meet  $a' - a \geq 0, b' - b \geq 0$ .

(3) Continuity condition: in order to prevent interval matching by matching step size, two adjacent elements in path  $p, p_k = (a, b)$  and  $p_{k+1} = (a', b')$  need to meet  $a - a' \leq 1, b - b' \leq 1$ .

After the DTW algorithm meets the above constraints, the optimal path is the minimum value of the cumulative sum of local distances  $DTW(\alpha, \beta)$ . The distance of DTW under the optimal path is shown in Eq. (15).

$$DTW(\alpha, \beta) = \min \left\{ \sum_{l=1}^L d(x_{ml}, y_{nl}) \right\} \tag{15}$$

The transformed time series  $\alpha', \beta'$  are expressed as:

$$\begin{cases} \beta' = \beta(pt(n)) \\ \alpha' = \alpha(pt(m)) \end{cases}, t = 1, 2, \dots, k \tag{16}$$

To address the problem of identical trend outputs when using real-time prediction models, which leads to confusion in single-feature decisions, additional features are introduced to improve the accuracy of the model. Root mean square can accurately characterize the time domain signal. The lower-limb behavior feature vector  $v$  of the real-time predictive mixture model is obtained as in Eq. (17).

$$v = [rms_{ac}, dtw] \tag{17}$$

**2.4. Establish abnormal gait trend monitoring methods**

In the initial stage of abnormal gait generated by the help robot, the recognition features under the point cloud distribution are difficult to recognize. When the user has abnormal gait in the middle and late stage of using the walking aid robot, the angle characteristics of lower-limb posture collected by point cloud data are difficult to distinguish and protect the user. Through the above analysis, in order to detect the abnormal gait trend, it can be transformed into a single classification problem of finite positive samples and infinite negative samples in the feature space, that is, to judge whether the lower-limb movement posture is normal walking.

In the selection of gait features for the real-time prediction model, the SVDD single classification algorithm [33] was used, and the samples selected were two time domain lower-limb behavior features. To satisfy the SVDD single classification model, a closure curve was resolved in the two-dimensional sample space as the boundary of the positive samples.

Let the positive sample vector set  $M = [v_1, v_2, v_3 \dots, v_m]$ , and the positive samples are distributed in the sphere with center  $a$  and radius  $r$ . The SVDD optimization objective function and optimization conditions are:

$$\begin{aligned} \min_{a, \xi_i} & \left( r^2 + C \sum_{i=1}^m \xi_i \right) \\ \text{s.t.} & \|v_i - a\|^2 \leq r^2 + \xi_i \\ & \xi_i \geq 0, i = 1, 2 \dots, m \end{aligned} \tag{18}$$

Parameter C is a penalty parameter, which is used to balance the sphere volume and misclassification rate in the sample space. Lagrange multiplier  $L(r, a, \alpha, \xi, \gamma)$  is introduced to describe the optimization objective:

$$L(r, a, \alpha, \xi, \gamma) = r^2 + C \sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i (r^2 + \xi_i - (\|v_i\|^2 - 2av_i + \|a\|^2)) - \sum_{i=1}^m \gamma_i \xi_i \tag{19}$$

In Eq. (18)  $\alpha_i \geq 0, \gamma_i \geq 0$ , let the partial derivatives of Lagrange function  $r$  and  $a$  and  $\xi_i$  be 0 respectively; then:

$$\begin{cases} \sum_{i=1}^m \alpha_i = 1 \\ a = \sum_{i=1}^m \alpha_i v_i \\ C - \alpha_i - \gamma_i = 0 \end{cases} \tag{20}$$

Substituting Eq. (20) into Lagrange function, the dual problem of SVDD is obtained as follows:

$$\max_a L(a) = \sum_{i=1}^m \alpha_i K(v_i, v_i) - \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j K(v_i, v_j) \tag{21}$$

The Lagrange coefficient of the sample can be obtained, and its rules are as follows:

$$\|v_i - a\|^2 < r^2 \rightarrow \alpha_i = 0, \gamma_i = 0 \tag{22}$$

$$\|v_i - a\|^2 = r^2 \rightarrow 0 < \alpha_i < C, \gamma_i = 0 \tag{23}$$

$$\|v_i - a\|^2 > r^2 \rightarrow \alpha_i = C, \gamma_i > 0 \tag{24}$$

The sample vector satisfying the condition (23) is called support vector. If  $v_s \in SV$ , then:

$$r^2 = K(v_s, v_s) - 2 \sum_{i=1}^n \alpha_i K(v_s, x_i) + \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K(v_i, v_j) \tag{25}$$

The distance  $d$  between the test sample  $v_t$  and the spherical center of the hypersphere is as follows:

$$d^2 = K(v_t, v_t) - 2 \sum_{i=1}^n \alpha_i K(v_t, v_i) + \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K(v_i, v_j) \tag{26}$$

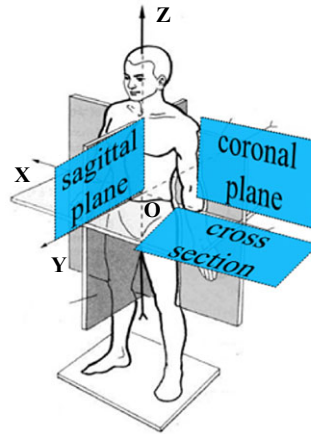


Figure 8. Three planes of human motion.

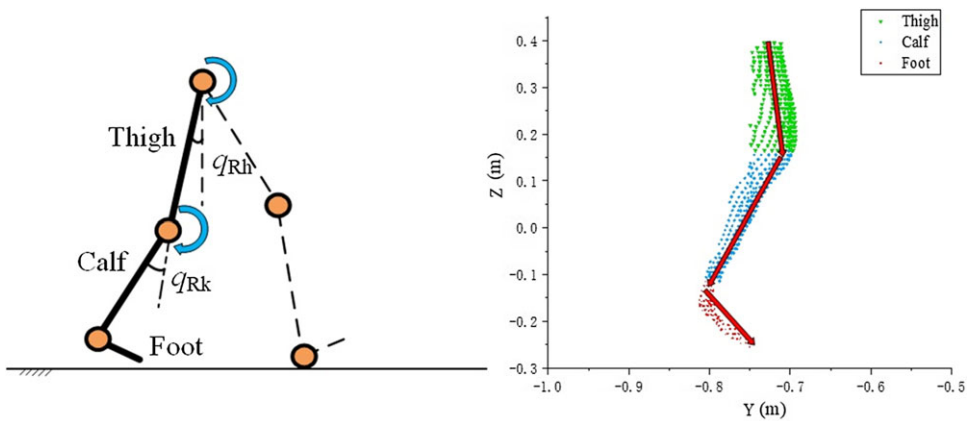


Figure 9. Lower-limb posture angle extraction in sagittal plane.

### 2.5. Establish the prediction model of lower-limb walking posture angle

Before establishing the lower-limb gait angle prediction model, the gait angle is selected. According to the mechanism of human anatomy, the horizontal, sagittal, and coronal views of human body are established, as shown in Fig. 8.

In this experiment, the sagittal plane of human anatomy is used to extract the posture angle of lower limbs. Since the depth camera captures three-dimensional data of the lower limbs, even after filtering, the point cloud data retains three-dimensional coordinate information of the hip, knee, and ankle joints. Therefore, in the event of the user leaning forward or backward abnormally, the point cloud coordinates can still be used to calculate the angles of the hip and knee joints. In this study, the depth camera was solely utilized to capture frontal 3D point cloud data of the lower limbs without considering user rotation. Due to occlusion issues encountered when obtaining specific values of each joint angle of the lower limbs in the coronal and transverse planes using the depth camera, point cloud data collection is restricted to the sagittal plane. The YOZ plane in the point cloud coordinate system is utilized as a projection plane within the anatomical sagittal plane to extract the posture angles of the lower limbs. Fig. 9 shows the projection of the right leg on the YOZ plane of the point cloud diagram. After all the lower limbs are projected to the YOZ plane, it can be regarded as the lower-limb-connecting rod model projected in the sagittal plane.

**Table II.** Lower-limb joint range of motion.

	Normal angle range	Maximum range of movement angles
$q_{RH}$	$-30^\circ \sim 145^\circ$	$-20^\circ \sim 75^\circ$
$q_{RK}$	$-135^\circ \sim 10^\circ$	$-65^\circ \sim 0^\circ$

As shown in Fig. 9,  $q_{RH}$  is the angle between the thigh and the Z-axis in the YOZ plane, and  $q_{RK}$  is the angle between the extension of the thigh linkage in the calf direction and the calf linkage. The range of specific activity angles is shown in Table 2.

After selecting the gait angle, the volunteers pushed forward under the conditions of indoor light and flat road and set the walking aid robot to the power-assisted working mode of moving forward at a constant speed. Each group of experiments pushed forward 2 m. According to the measurement, a gait cycle lasts about 2.5 s when moving forward slowly. In addition, depth images are collected at a rate of 60 frames per gait cycle to obtain the complete motion walking cycle. Due to four types of falls, each repeated 10 times, a total of 6000 depth visual images were collected for each person, totaling  $60 * 2.5 * 4 * 10$ . Fifteen people collected a total of 90,000 depth images, each with 4 joint angles, resulting in a total of 360000 joint point data.

### 2.5.1. Dataset and sample set construction

In this study, LSTM network was utilized to predict lower-limb posture angles. A dataset comprising a total of 30,000 motion posture angle images of bilateral hip and knee joints in the sagittal plane was selected from five subjects for deep learning training. Of these, 80% of the data was allocated to the training set, while the remaining 20% was used for validation.

The training model takes the right leg knee angle as an example. Starting from the first data of the right leg knee angle dataset, take the first  $j$  frame data as the input of the sample vector and the  $j + i$  frame data as the training label of the sample vector; that is, the first sample vector is  $X_{Lh}^1 = [x_{Lh}^1, x_{Lh}^2, x_{Lh}^3, \dots, x_{Lh}^j]$ , the label value  $Y_{Lh}^1 = x_{Lh}^{j+i}$ , and the lower-limb knee angle dataset takes the sliding window value for each frame of the picture until the sliding window moves to the end of the dataset. Since the sampling frequency of the depth camera is 60 fps (frame per second), the conversion relationship between the predicted step width of the sample ( $n$ ) and the predicted time step ( $t_{predict}$ ) can be calculated as shown in Eq. (27).

$$t_{predict} = \frac{1}{60}n \quad (27)$$

The deep learning model is trained using a linear function normalization method to calculate the updated weight matrix, such that the training set accounts for sixty percent of the total deep learning dataset, the validation set is twenty percent of the total deep learning dataset, and the test set is twenty percent of the total deep learning dataset.

### 2.5.2. Build a deep learning network model

The training goal of the deep learning model designed in this study is to predict the motion information of four joints of the lower limb. There are four units in the input layer and output layer of the deep learning network structure. The input layer parameter is that each input unit has a one-dimensional vector corresponding to it, that is, each lower-limb joint angle  $x_i$ , and the step size of each sampling is set to  $k$  frames.  $k$  is determined by the frame rate of the depth camera with various processor computing power. Each element in the four sample label vectors  $y_i$  is used as the training label of the corresponding input unit. The format of the output layer is a single-layer fully connected neural network for linear transformation of the feature space where the hidden layer is located to the label space. In consideration of the principle of lightweight and practicality of the network model, the number of hidden layers is chosen as three layers, all of which are LSTM cell layers. The hidden layer consists of multiple LSTM

**Table III.** LSTM prediction model training parameters.

Model parameter	Parameter selection
Loss function	MSE
Optimizer	Adam
Neuron inactivation ratio	0.2
LSTM layers	3
Number of cell states in each layer of LSTM	100, 80, 60
Number of nodes in the whole connection layer	10
Enter the number of layer nodes	4
Learning rate	0.000001
Number of iterations	300

cells and connects the input layer with the output layer. In this deep learning model, the activation function of LSTM unit layer is tanh function (28), the activation function of output full link layer unit is sigmoid function (29), and the loss function is MSE (30).

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (28)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (29)$$

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^m (y_i - \tilde{y}_i)^2 \quad (30)$$

The software used for deep learning model training includes Python 3.8, Keras 2.4.2, Numpy 1.18.4, and Scikitlearn 0.23.2. The hardware used for deep learning model training includes Intel i7 processor and RTX4000 graphics card. After considering the parameters of the mainstream deep learning network training model at this stage and training the deep learning network several times, the final parameters of the deep learning network model in this study are set as shown in Table 3. In the deep learning training model, the sagittal angles of four joints of both lower limbs are selected as the dataset for training. In order to improve the generalization ability of the model, the time series data of participants under different walking speeds and asynchronous amplitudes collected during the experiment were mixed, and, finally, 10,000 frames of continuous change data of effective walking posture angles of four joints were sorted out.

### 3. Results

#### 3.1. Prediction model experiment and results

According to the selected deep learning training model, the relevant information of four lower-limb joint movements in a gait cycle is randomly selected from the verification results of the verification set. The original data and prediction data are shown in Fig. 10.

As can be seen from Fig. 10, according to the comparison diagram between the test dataset and the actual prediction data, the prediction data can accurately predict the test data in 0.2 s, so the prediction ability of this model is strong.

In this experiment, three evaluation criteria are cited to describe the error between the label value and the model output in the test sample, which are mean absolute error (MAE), mean square error (MSE), and root mean square error (RMSE). To verify the prediction ability of the LSTM network, three evaluation criteria were used to compare the prediction ability performed with the Back Propagation

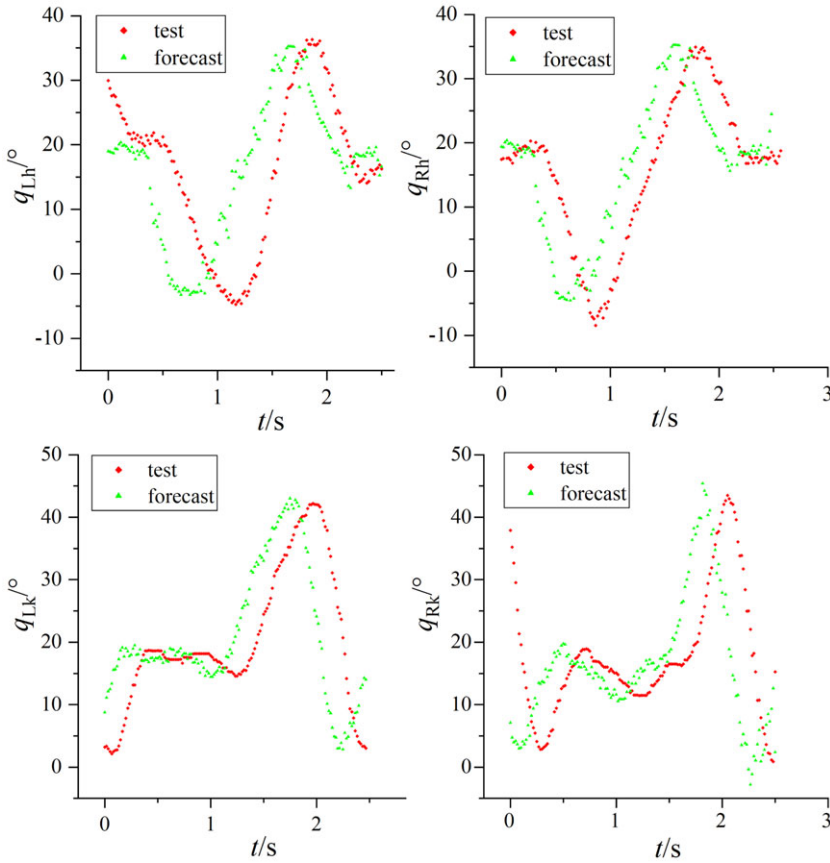


Figure 10. LSTM prediction model test results.

neural network (BP) and Recursive Neural Network (RNN) network and Convolutional Neural Networks (CNN) network and Gated Recurrent Units (GRUs) and Bidirectional Encoder Representations from Transformers (BERT) with the equal number of layers, the sizes of these six training models are also compared, and the results are shown in Table 4.

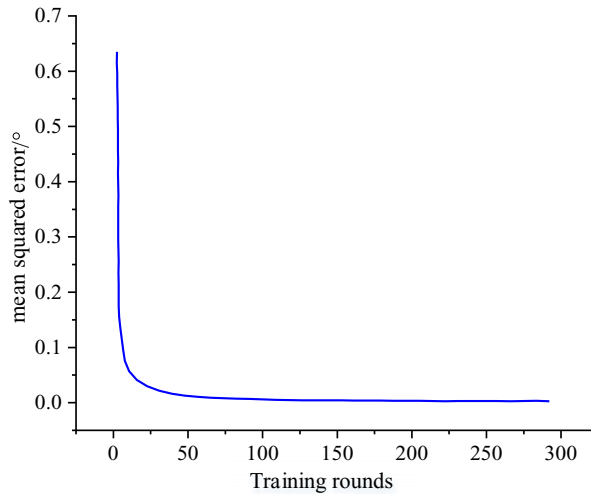
As shown in Table 4, in the test set samples, the LSTM network prediction model exhibits smaller MAE, MSE, and RMSE values compared to other models, demonstrating its superior predictive accuracy. Although the trained models of BP and RNN are smaller than the LSTM model, their predictive capabilities are weaker, making them unsuitable as predictive models. Both the CNN and GRU models demonstrate predictive capabilities close to the LSTM model, but they are larger in size. While the predictive capabilities of the BERT model are comparable to those of the LSTM model, the BERT model is significantly larger. Considering the limited computational resources of the control module, the LSTM model is chosen as the predictive model. Taking into account both the predictive capability and the size of the model, LSTM is selected as the predictive model.

Figure 11 shows the variation curve of mean square error for the loss function during the training process of the long short-term memory network model.

As shown in Fig. 11, with the increase of training rounds the loss function tends to be smooth, and finally, the model tends to converge, proving that the model is reliable and effective. Table 5 shows the specific height, weight, age, number of experiments and the accuracy of the validation set predicted using the LSTM model for the five subjects.

**Table IV.** Evaluation and comparison of multi-model prediction results.

Network model	MAE	MSE	RMSE	SIZE(M)
BP	$1.6598 \times 10^{-1}$	$9.7461 \times 10^{-3}$	$8.6522 \times 10^{-2}$	9.25
RNN	$8.6579 \times 10^{-2}$	$6.6978 \times 10^{-3}$	$5.3659 \times 10^{-2}$	12.78
CNN	$6.4763 \times 10^{-2}$	$5.5972 \times 10^{-3}$	$4.1549 \times 10^{-2}$	16.36
GRU	$5.8215 \times 10^{-2}$	$4.2941 \times 10^{-3}$	$3.6426 \times 10^{-2}$	15.59
BERT	$4.6732 \times 10^{-2}$	$4.0568 \times 10^{-3}$	$2.9713 \times 10^{-2}$	19.32
LSTM	$4.2691 \times 10^{-2}$	$3.8542 \times 10^{-3}$	$2.5985 \times 10^{-2}$	13.17



**Figure 11.** Variation of loss function values during model training.

**3.2. Abnormal gait monitoring and performance analysis of lower limbs**

When the user stands or assists walking with the walker robot, the depth camera monitors the trend of the user’s lower-limb behavior, thus preventing the user from leaning forward, falling sideways, or falling. A block diagram of the safety performance enhancement scheme of this study is shown in Fig. 12.

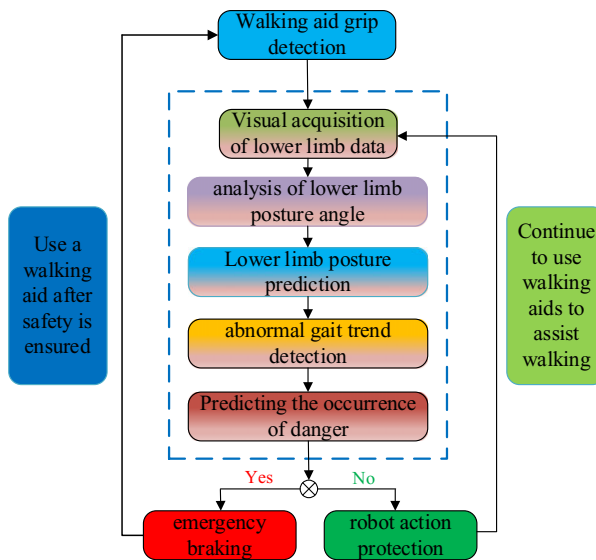
When users engage the mobility aid, deep vision technology is employed to extract point cloud data for monitoring the pose of the user’s lower-limb joints. This involves employing an offline-trained angle prediction model to analyze the angles of the lower-limb joints. Subsequently, prediction of the pose angles is accomplished using decision boundaries based on abnormal gait trends. Initial monitoring of abnormal walking trends is conducted, integrating the empirical trajectories of key points from a dynamic model for the modal recognition of walking abnormalities. The system ultimately determines whether there is an occurrence of abnormal gait trends. In the event of predicting a potential danger, the mobility aid initiates an emergency brake to prevent any harm to the user. If no dangerous gait trends are detected, the aid continues to assist the user in rehabilitation training.

The dynamic time programming algorithm judges the tendency for abnormal lower-limb movements when the boundary conditions, monotonicity conditions, and continuity conditions are met, and the optimal path is obtained as shown in Fig. 13.

From Eq. 25, when  $d \leq r$ , the test sample is inside or on the boundary of the positive sample, which is regarded as a normal sample; when  $d > r$ , the test sample is outside the boundary and is regarded as an abnormal sample. Take the angle data of left knee joint as an example. Let the function be the model training kernel function, establish the normal walking positive sample set, and take the normal lower-limb attitude angle sampling experimental data as the total dataset, the window rule of online monitoring

**Table V.** Detailed parameters and validation set accuracy of five subjects.

Gender	Height (cm)	Weight (kg)	Age	Repetitions	Accuracy
Female	165.1	53.1	26	10	98.6%
Female	163.4	56.2	23	10	96.4%
Male	176.8	75.9	25	10	98.1%
Male	175.6	77.8	26	10	96.9%
Male	178.5	78.9	25	10	98.3%

**Figure 12.** Safety performance improvement scheme.

is that the width is 40 frames and the step length is three frames, and 600 groups of continuous positive training samples are obtained, and the realistic data of the real-time prediction model is compared with the predicted data as shown in Fig. 14.

When training with the Gaussian kernel function, the parameter  $s$  needs to be tuned. It is found through the experiment that the degree of fitting of the decision boundary is negatively correlated with  $s$ , and the model has the best decision effect when  $s$  is 200. The training results are shown in Fig. 15.

As shown in Fig. 15(a), when  $s$  is set to 200, most training samples fall within the decision boundary of 0.885, with only a small number of samples lying outside the boundary. This indicates that the kernel function exhibits a good level of fit when  $s$  is 200, without displaying signs of overfitting. Additionally, as illustrated in Fig. 15(b), the distribution of sample distances predominantly ranges between 0.84 and 0.89, suggesting that the decision boundary performs optimally at this range, thereby providing favorable boundary conditions for subsequent prediction of lower-limb gait risks.

### 3.3. Abnormal gait trend monitoring results

Since the subjects are performing normal walking recovery training with lower-limb walking aids, when the patients encounter an emergency during walking at a uniform speed suitable for them, the abnormal gait trend monitoring will quickly monitor and judge the abnormality within the current gait cycle, as shown in Fig. 16.

Figure 16(a) shows the test data and prediction data of an abnormal attitude angle of the left knee. An exception occurs when it is about 2.5 s. The width of the feature sampling window is 500 ms and the



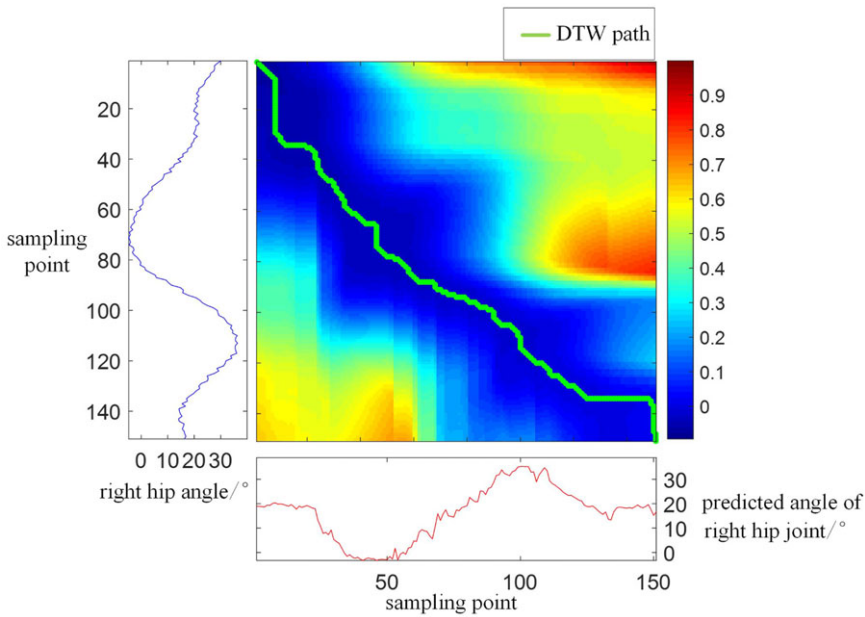


Figure 13. The optimal path of DTW.

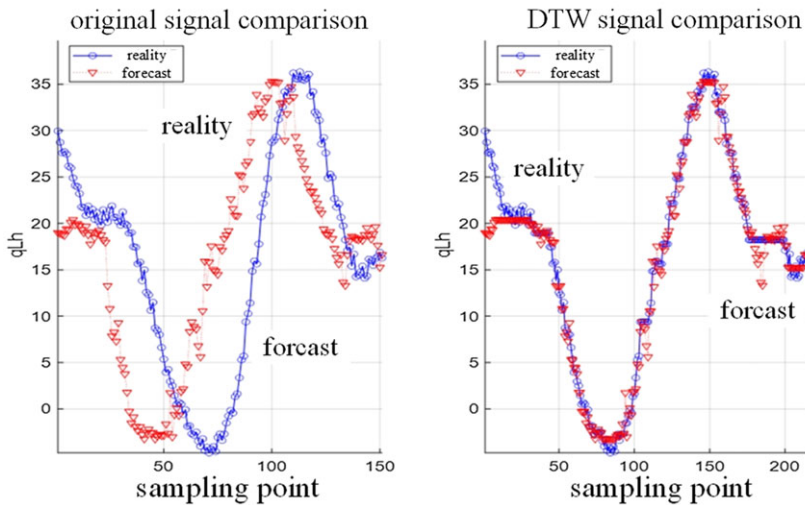
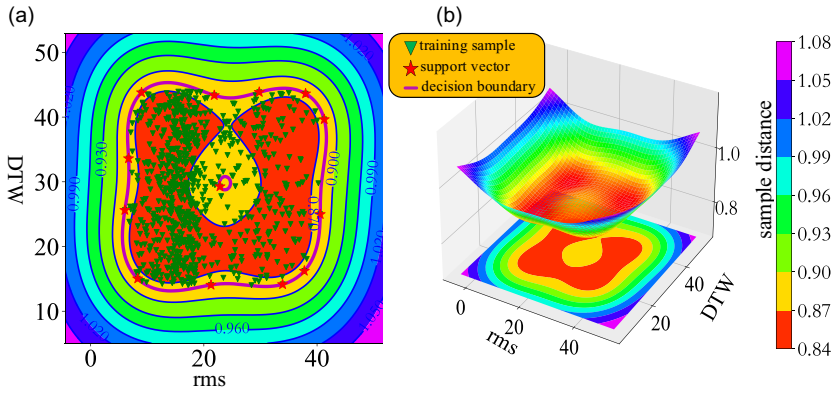


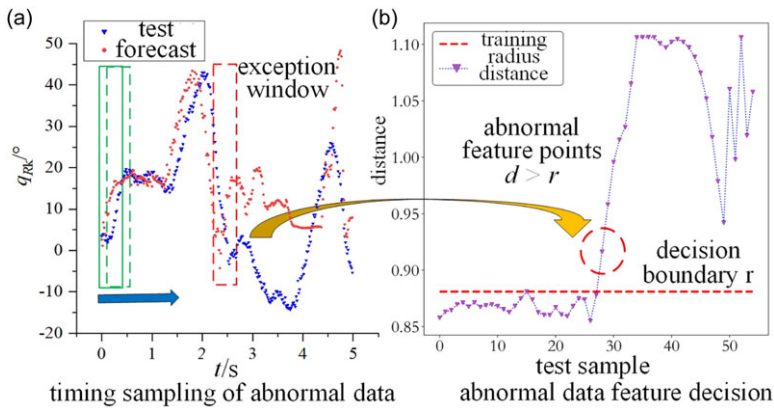
Figure 14. Comparison of attitude angle information of real-time prediction model using DTW.

step size is 60 ms. In Fig. 16(b), the horizontal axis is the sliding window sequence, the vertical axis is the distance  $d$  between the current sample and the center of the decision model, and the red dotted line represents the decision boundary  $r$  of the model. In the decision model, an anomaly is detected in the 34th sampling window, and a continuous anomaly decision is made in the subsequent sampling window. This experiment verifies the sensitivity of the real-time prediction model to abnormal gait and the good real-time response ability of the single classification decision model in abnormal gait monitoring.

To further validate the real-time prediction model's real-time and accuracy, a mixed sample was created to evaluate the model. Ten time series posture angle data with a global time window of 2.5 s, including the initial abnormal gait trend, were selected for testing, and the final results were 396 positive



**Figure 15.** Decision boundary training results based on Gaussian kernel function: (a) Two-dimensional decision boundary of training samples; (b) Sample distance distribution.



**Figure 16.** Online abnormal trend monitoring scheme.

samples and 199 negative samples. Decisions with five or more consecutive exceptions were specified as valid decisions. The decision delay is the difference between the actual anomaly time point and the model decision time point. A negative number indicates that anomalies can be predicted. The test results are shown in Table 6. In order to further verify the real-time and accuracy of the real-time prediction model, the remaining ten subjects were selected for data collection, and mixed samples were established to evaluate the model.

From Table 6, it is clear that the real-time prediction model can accurately determine the abnormalities of hip and knee posture angles; however, the probability of misclassification is higher for positive samples from the lower-limb knee than the hip. The abnormal decision of the knee joint preceded that of the hip joint, which indicated that the knee joint was better in real time. The reason for this is that the range of motion of the knee joint posture angle is larger than that of the hip joint posture angle, so the model is more resistant to interference and has better tracking ability.

To validate the model’s generalization ability, the remaining ten participants’ lower-limb depth vision data were selected for prediction verification. These data from the ten participants were not included in the training of the LSTM prediction model. Table 7 provides specific information and the accuracy of the model validation for the ten participants.

From Table 7, it can be observed that the LSTM prediction model achieves an overall accuracy of over 95%, with an average prediction accuracy of 96.33% across the ten participants. This indicates a high level of predictive capability.

**Table VI.** Anomaly monitoring model evaluation.

Attitude angle category	Positive sample recognition rate	Positive sample confusion rate	Negative sample recognition rate	Negative sample confusion rate	Average abnormal decision delay
q <sub>Lh</sub>	91.2%	8.8%	96.3%	3.7%	20 ms
q <sub>Rh</sub>	90.7%	9.3%	97.6%	2.4%	13 ms
q <sub>Lk</sub>	88.8%	11.2%	98.2%	1.8%	– 131 ms
q <sub>Rk</sub>	87.5%	12.5%	97.5%	2.5%	– 122 ms

**Table VII.** Detailed parameters and model prediction accuracy for ten subjects.

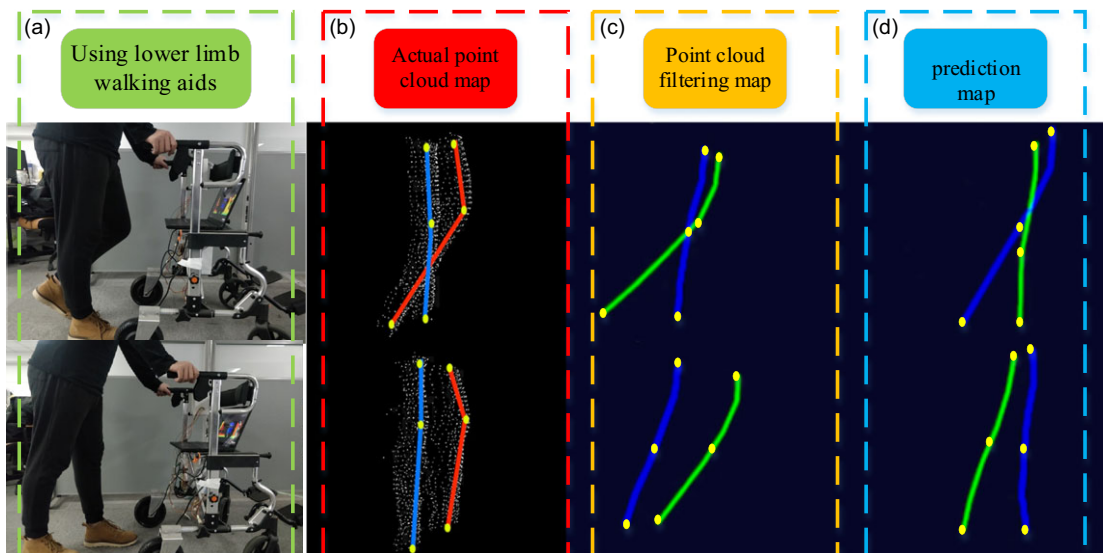
Gender	Height (cm)	Weight (kg)	Age	Repetitions	Accuracy
Female	169.6	55.3	24	10	98.2%
Female	166.9	56.9	25	10	97.3%
Female	163.2	58.2	25	10	94.7%
Female	165.4	57.1	23	10	95.8%
Male	172.5	68.5	25	10	96.5%
Male	173.5	71.5	24	10	96.2%
Male	178.2	80.4	23	10	97.5%
Male	175.3	75.9	25	10	96.1%
Male	176.2	77.2	24	10	95.6%
Male	179.2	82.9	23	10	95.4%

Figure 17 depicts the actual results at two randomly selected time points in lower-limb prediction using the LSTM prediction model. Figure 17(a) illustrates the lower-limb movements of the user while employing the lower-limb assistive walker. Figure 17(b) depicts the number of point clouds in the lower limbs at the same moment of walker usage. Figure 17(c) showcases the filtered lower-limb point cloud data, where redundant information has been removed while still retaining essential lower-limb data. Figure 17(d) demonstrates the real-time prediction of lower-limb actions based on the filtered lower-limb point cloud data. The actual performance depicted in Fig. 17(d) indicates that the LSTM prediction model possesses robust capabilities in forecasting lower-limb risky behaviors.

#### 4. Discussion

The research findings indicate that the LSTM model can accurately predict test data within 0.2 s during randomly selected gait cycles, showcasing its strong predictive speed capability. Compared to other deep learning models, the LSTM model demonstrates higher prediction accuracy than BP and RNN models. In comparison to the GRU model, the LSTM model outperforms in both prediction accuracy and model size. Although the BERT model shows similar prediction accuracy to the LSTM model, it is larger by 6.15M compared to the LSTM model. Therefore, the LSTM model exhibits the strongest overall capability. As the training epochs of the LSTM model reach 150 rounds, the loss function tends to smooth out, and the final model converges. In the LSTM model’s prediction validation set of 5 participants, the average accuracy reaches 97.66%, confirming the reliability and effectiveness of the model.

Meanwhile, DTW allows for obtaining the optimal path for the four lower-limb joints. The optimal path derived from DTW effectively reduces the time gap between reality and prediction, enabling the training model to quickly identify abnormal behaviors and enhance the safety of pedestrians. The average recognition rate for positive samples of the four lower-limb joints is 89.55%, with a tolerance of 3.7%. For negative samples of the four lower-limb joints, the average recognition rate is 97.4%, with a tolerance



**Figure 17.** Actual effect diagram of LSTM prediction of lower-limb behavior.

of 1.9%. These results indicate that the prediction model can accurately detect abnormal lower-limb behaviors.

In the validation testing to assess the model's generalization capability, the LSTM prediction model achieved an overall accuracy of over 95% for the 10 subjects, with the highest accuracy reaching 98.2%. The average prediction accuracy was 96.33%. Moreover, it demonstrated strong predictive capability in practical applications. In conclusion, this real-time prediction model exhibits excellent real-time performance and accuracy, making it suitable for rapid fall risk alerting.

A deficiency of this study is that, because the user needs rehabilitation training with a lower-limb walker, this study predicts the abnormal gait in the same gait cycle when the user advances at a uniform speed but does not predict the behavior of the user at different walking speeds. In terms of future work, the experimental team and I will continue to conduct in-depth research to solve the problem of predicting the abnormal lower-limb behavior of users under different gait cycles and variable speeds of walking.

## 5. Conclusion

This study proposed a prediction model of human lower-limb abnormal behavior based on 3D vision. Firstly, a real-time lower-limb posture angle resolution method based on point cloud data was designed, and the lower-limb information was extracted accurately and quickly using multilayer filters; secondly, a deep learning model for simultaneous prediction of multiple posture angle falls was designed based on the periodic characteristics of walking motion posture angles in the sagittal plane, and a method for characterizing real-time walking motion states was proposed; finally, the real-time and accuracy of the real-time prediction model was verified using mixed samples, and the prediction model was sufficient to make accurate prediction of abnormal gait within 150 ms. In general, this study uses point cloud data to extract the behavior characteristics of lower limbs and predict the dangerous behavior quickly and accurately, which puts forward a new engineering technology method for improving the safety of a lower-limb walking aid robot.

In future work, a human-machine coupled control system can be established based on this prediction model to further improve the safety performance of the walking aid robot.

**Author contributions.** The first author Tie Liu wrote this article, the second author Dianchun Bai and the fourth author Hiroshi Yokoi conceived and designed this study, and the third author Hongyu Yi collected and analyzed the data of this study.

**Financial support.** This work has received partial support from the Ministry of Education's Chunhui Plan 202200215 funding.

**Competing interests.** The authors declare no conflicts of interest exist.

**Ethical approval.** Not applicable.

## References

- [1] B. Chen, H. Ma, L.-Y. Qin, F. Gao, K.-M. Chan, S.-W. Law, L. Qin and W.-H. Liao, "Recent developments and challenges of lower extremity exoskeletons," *J Orthop Transl* **5**, 26–37 (2016).
- [2] Y.-L. Wang, K.-Y. Wang, W.-Y. Zhao, W.-L. Wang, Z. Han, & Z.-X. Zhang, "Effects of single crouch walking gaits on fatigue damages of lower extremity main muscles," *J Mech Med Biol* **19**(07), 1940046 (2019).
- [3] E. Godecke, E. Armstrong, T. Rai, N. Ciccone, M. L. Rose, S. Middleton, A. Whitworth, A. Holland, F. Ellery, G. J. Hankey, D. A. Cadilhac and J. Bernhardt, "A randomized control trial of intensive aphasia therapy after acute stroke: The very early rehabilitation for spEech (VERSE) study," *Int J Stroke* **16**(5), 556–572 (2020).
- [4] H.-C. Yen, J.-S. Jeng, W.-S. Chen, G.-S. Pan, W.-Y. Chuang, Y.-Y. Lee and T. Teng, "Early mobilization of mild-moderate intracerebral hemorrhage patients in a stroke center: A randomized controlled trial," *Neurorehabil Neural Repair* **34**(1), 72–81 (2019).
- [5] Q. Yan, J. Huang, C. Tao, X. Chen and W. Xu, "Intelligent mobile walking-aids: Perception, control and safety," *Adv Robotics* **34**(1), 2–18 (2020).
- [6] K. Ominato and T. Murakami, "A Stabilization Control in Two-Wheeled Walker with Passive Mechanism for Walking Support, IECON," **In: 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society**, (2020) pp. 65–70.
- [7] E. L. Oyman, M. Y. Korkut, C. Yilmaz, Z. Y. Bayraktaroglu and M. S. Arslan, "Design and control of a cable-driven rehabilitation robot for upper and lower limbs," *Robot* **40**(1), 1–37 (2022).
- [8] Y. Ye, M.-X. Zhu, C.-W. Ou, B.-Z. Wang, L. Wang and N.-G. Xie, "Online pattern recognition of lower limb movements based on sEMG signals and its application in real-time rehabilitation training," *Robotica* **42**(2), 389–414 (2024).
- [9] Y.-L. Wang, K.-Y. Wang, Y.-J. Chai, Z.-J. Mo and K.-C. Wang, "Research on mechanical optimization methods of cable-driven lower limb rehabilitation robot," *Robotica* **40**(1), 154–169 (2022).
- [10] J.-H. Wang and J.-Y. Kim, "Development of a whole-body walking rehabilitation robot and power assistive method using EMG signals," *Intel Serv Robot* **16**(2), 139–153 (2023).
- [11] R. S. Gonçalves and L. A. O. Rodrigues, "Development of nonmotorized mechanisms for lower limb rehabilitation," *Robotica* **40**(1), 102–119 (2022).
- [12] S. H. Hwang, D. I. Sun, J. Han and W.-S. Kim, "Gait pattern generation algorithm for lower-extremity rehabilitation-exoskeleton robot considering wearer's condition," *Intel Serv Robot* **14**(3), 345–355 (2021).
- [13] C. Ophaswongse, R. C. Murray, V. Santamaria, Q. Wang and S. K. Agrawal, "Human evaluation of wheelchair robot for active postural support (WRAPS)," *Robotica* **37**(12), 2132–2146 (2019).
- [14] Y. Li, Y. Wang, S. Yuan and Y. Fei, "Design, modeling, and control of a novel soft-rigid knee joint robot for assisting motion," *Robotica* **42**(3), 817–832 (2024).
- [15] V. B. Semwal, R. Jain, P. Maheshwari and S. Khatwani, "Gait reference trajectory generation at different walking speeds using LSTM and CNN," *Multimed Tools Appl* **82**(21), 33401–33419 (2023).
- [16] S. Kumar, P. Yadav and V. B. Semwal, "A Comprehensive Analysis of Lower Extremity Based Gait Cycle Disorders and Muscle Analysis," **In: Machine Learning, Image Processing, Network Security and Data Sciences** (N. Khare, D. S. Tomar, M. K. Ahrwal, V. B. Semwal and V. Soni, eds.) (Springer Nature Switzerland, Cham, 2022) pp. 325–336.
- [17] P. Zhang, J. Zhang and A. Elsabbagh, "Fuzzy radial-based impedance controller design for lower limb exoskeleton robot," *Robotica* **41**(1), 326–345 (2023).
- [18] W. Yuqi, C. Jinjiang, G. Ranran, Z. Lei and W. Lei, "Study on the design and control method of a wire-driven waist rehabilitation training parallel robot," *Robotica* **40**(10), 3499–3513 (2022).
- [19] G. Ercolano and S. Rossi, "Combining CNN and LSTM for activity of daily living recognition with a 3D matrix skeleton representation," *Intel Serv Robot* **14**(2), 175–185 (2021).
- [20] T. Qin, Y. Yang, B. Wen, Z. Chen, Z. Bao, H. Dong, K. Dou and C. Yang, "Research on human gait prediction and recognition algorithm of lower limb-assisted exoskeleton robot," *Intel Serv Robot* **14**(3), 445–457 (2021).
- [21] S. K. Challa, A. Kumar, V. B. Semwal and N. Dua, "An optimized-LSTM and RGB-D sensor-based human gait trajectory generator for bipedal robot walking," *IEEE Sens J* **22**(24), 24352–24363 (2022).
- [22] Y. Lou, R. Wang, J. Mai, N. Wang and Q. Wang, "IMU-based gait phase recognition for stroke survivors," *Robotica* **37**(12), 2195–2208 (2019).
- [23] V. B. Semwal, A. Mazumdar, A. Jha, N. Gaud and V. Bijalwan, Speed, Cloth and Pose Invariant Gait Recognition-Based Person Identification," **In: Machine Learning: Theoretical Foundations and Practical Applications** (M. Pandey and S. S. Rautaray, eds.) (Springer Singapore, Singapore, 2021) pp. 39–56.

- [24] V. B. Semwal, Y. Kim, V. Bijalwan, A. Verma, G. Singh, N. Gaud, H. Baek and A. M. Khan, “Development of the LSTM model and universal polynomial equation for all the sub-phases of human gait,” *IEEE Sens J* **23**(14), 15892–15900 (2023).
- [25] S. Gaglio, G. L. Re and M. Morana, “Human activity recognition process using 3-D posture data,” *IEEE Trans Hum-Mach Syst* **45**(5), 586–597 (2015).
- [26] W. Xu, J. Huang and L. Cheng, “A novel coordinated motion fusion-based walking-aid robot system,” *Sensors* **18**(9), 2761 (2018).
- [27] W. Xu, D. Xiang, G. Wang, R. Liao, M. Shao and K. Li, “Multiview video-based 3-D pose estimation of patients in computer-assisted rehabilitation environment (CAREN),” *IEEE Trans Hum-Mach Syst* **52**(2), 196–206 (2022).
- [28] J. White, T. Kameneva and C. McCarthy, “Vision processing for assistive vision: A deep reinforcement learning approach,” *IEEE Trans Hum-Mach Syst* **52**(1), 123–133 (2022).
- [29] L. Yang, Y. Ren and W. Zhang, “3D depth image analysis for indoor fall detection of elderly people,” *Digit Commun Netw* **2**(1), 24–34 (2016).
- [30] C. K. Lee and V. Y. Lee, “Fall Detection System Based on Kinect Sensor Using Novel Detection and Posture Recognition Algorithm,” *In: Inclusive Society: Health and Wellbeing in the Community, and Care at Home* (J. Biswas, H. Kobayashi, L. Wong, B. Abdulrazak and M. Mokhtari, eds.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2013) pp. 238–244.
- [31] A. Gupta and V. B. Semwal, “Occluded gait reconstruction in multi person gait environment using different numerical methods,” *Multimed Tools Appl* **81**(16), 23421–23448 (2022).
- [32] J. Stübinger and D. Walter, “Using multi-dimensional dynamic time warping to identify time-varying lead-lag relationships,” *Sensors* **22**(18), 6884 (2022).
- [33] D. M. J. Tax and R. P. W. Duin, “Support vector data description,” *Mach Learn* **54**(1), 45–66 (2004).