Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 9536309, 10 pages
https://doi.org/10.1155/2021/9536309

Research Article

An Intelligent Optimization Strategy Based on Deep Reinforcement Learning for Step Counting

Zhoubao Sun ,[1] Pengfei Chen,[2] and Xiaodong Zhang[1]

[1]*Jiangsu Key Laboratory of Public Project Audit, Nanjing Audit University, Nanjing 211815, China*
[2]*Department of Economics, University of California, San Diego 92093-0508, CA, USA*

Correspondence should be addressed to Zhoubao Sun; 270247@nau.edu.cn

Received 31 July 2021; Accepted 18 November 2021; Published 7 December 2021

Academic Editor: Juan L. G. Guirao

With the popularity of Internet of things technology and intelligent devices, the application prospect of accurate step counting has gained more and more attention. To solve the problems that the existing algorithms use threshold to filter noise, and the parameters cannot be updated in time, an intelligent optimization strategy based on deep reinforcement learning is proposed. In this study, the counting problem is transformed into a serialization decision optimization. This study integrates the noise recognition and the user feedback to update parameters. The end-to-end processing is direct, which alleviates the inaccuracy of step counting in the follow-up step counting module caused by the inaccuracy of noise filtering in the two-stage processing and makes the model parameters continuously updated. Finally, the experimental results show that the proposed model achieves superior performance to existing approaches.

1. Introduction

Accurate and efficient step counting plays an important role in exercise [1]. With the development of Internet of things (IoT) and artificial intelligence technology, a variety of motion monitoring devices have sprung up, such as pedometer and sports bracelet.

As a method of motion statistics, step counting is achieved by analyzing the gait characteristics, which mainly refer to the individual behavior characteristics of human body in normal walking, including speed, amplitude, and posture, and different individuals with strong differences [2]. There are mainly two kinds of step counting devices, one is customized pedometer, which has single function and stable effect, and needs to fix special hardware equipment [3]. The other is common applications that are based on gait recognition system and are relatively simple, feasible, and easy to implement to obtain motion data through acceleration sensors.

Step counting has been widely used in personal health assistant, medical health monitoring, positioning navigation, etc. With the development of IoT, the step counting equipment can achieve a good accuracy rate, the function of the sensor is more powerful, the data acquisition is more convenient and accurate, and the anti-interference ability is stronger. However, there are two shortcomings that need to be seriously taken into consideration: one is the user's various behaviors that are easy to cause noise, such as arm shaking, steering, squatting, and falling; the other is the parameters that cannot be continuously optimized. The traditional pedometer mostly uses signal processing to analyze the sampled data and count the steps. However, different users have different uncertain behaviors in the actual environment, so the noise data are generated randomly. The general approach is to add threshold to filter the noise data. How to optimize the threshold value brings new problems, in practice, the high error.

How to solve the abovementioned problems? In this study, an intelligent optimization algorithm based on deep reinforcement learning [4] is proposed, which overcomes the shortcomings of parameter updating, optimization, and self-learning when filtering noise data. Therefore, the original problem is transformed into a series of decision optimization. In the deep reinforcement learning, with the representation learning ability of deep learning, the agent can automatically and autonomously learn the effective features from the original data. Experiments on real data sets verify that the proposed model could optimize the ability of noise recognition and improve the final accuracy.

The remainder of the study is organized as follows: the related work is presented in Section 2; the problem and the details of the proposed approach are shown in Section 3; the experimental results are shown in Section 4; and finally, we draw the conclusion in Section 5.

2. Related Work

Recently, a variety of step counting applications have been developed, and more approaches are constantly proposed. Many literatures have done research on the aspect. The traditional approaches include threshold methods and wave peak detection, in which the embedded module is mainly used and the calculation ability is weak. With the development of microelectronics technology and the emergence of intelligent devices, the computing power improved, and more and more new type approaches have been proposed.

2.1. Traditional Approaches. Traditional approaches firstly collect the data of acceleration sensor, then smooth the waveform, and finally count the steps by signal processing approach.

The threshold method is used to set up various conditions through the curve change in acceleration waveform. If the change in signal waveform meets these conditions, it is recorded as 1. Zhang et al. [5] used the method of dynamic threshold to count steps; he calculates the mean value of the data waveform as the threshold value and constantly updates the threshold value with the progress of sampling. When the sampling point is less than the current threshold for the first time, it is recorded as a step. He also pointed out in the literature that this method has strong sensitivity. When the pedometer vibrates frequently or vibrates slowly but is not walking, the pedometer still regards it as a step. In Ref. [6], the sensor is tied to the ankle to count steps according to the acceleration whether exceeds a threshold or not. In the process of reckoning, the authors [7] combine the data of vertical angle with the autocorrelation coefficient method to calculate the step, which shows good detection accuracy, but it also has the problem of high calculation cost. Chen et al. [8] used the ensemble network architecture for deep reinforcement learning, which is based on value function approximation. Jayalath and Abhayasinghe [9] proposed to use a gyroscope as acquisition sensor to calculate the number of steps by calculating the number of times of crossing the zero point.

Peak detection approach counts the number of wave peaks in sensor data to achieve step counting. The disadvantage is that it is easy to be affected by pseudo-peaks. Lu and Velipasalar [10] used the peak detection method to detect the trough and then calculated the trough value. Lou et al. [11] combined the intermediate threshold method to determine the wave peak value. Lou et al. [12] proposed an adaptive peak detection method, and according to the set threshold, the normal state and abnormal state are determined, and then, different neighborhood windows are set for different states, and the wave peaks are counted in the window. Liu and Yang [13] designed an adaptive time window to determine the wave crest and trough simultaneously through the adaptive double threshold. The peak detection approach has less calculation and is easy to implement. It can effectively complete the step counting in simple scenarios, but it has weak ability to identify noise, especially when the acquisition equipment vibrates randomly, such as shaking.

In addition, waveform analysis is a kind of the similarity calculation approaches, which counting steps by analyzing the ascending and descending trends of acceleration data. Wang et al. [14] abstracted the change in waveform into three states: stationary state, peak state, and trough state. Tang et al. [15] calculated the horizontal acceleration, vertical acceleration, and angle from linear acceleration and gravity acceleration, and calculated the distance between the collected data and the wave data in the sampling set. Rai et al. [16] proposed an autocorrelation analysis approach, which calculates the similarity between the data collected in the previous period and the current period. The similarity calculation method can analyze the periodic data well, but it needs to analyze the cycle of the data in advance. For different speeds of walking, the data cycle would be different.

2.2. State-of-the-Art Methods. Due to the defects of traditional approaches, various improved ones are proposed. Xiao [17] found that the interference signal of the waist is not completely random and designed an approach by Freeman coding the signal; in Ref. [18, 19], the authors designed an approach that tests the signal waveform, which has three obvious phase changes, namely the downward swing phase, the upward swing phase, and the standing phase. The above two approaches are based on local features to improve the accuracy, with high time complexity and implementation complexity. However, the accuracy rate in practical application is not as high as that of the threshold method, especially in the case of long-time walking.

With the rapid development of intelligent devices, more and more approaches are proposed. In Ref. [20], the authors divided the movement into rhythmic and nonrhythmic activities by collecting the triaxial acceleration data of the wrist. When the oscillation meets the requirements of a certain length, intensity, and shape, it is considered as step counting data. In Ref. [21], the authors proposed to use a gyroscope as the acquisition sensor to calculate the number of steps by calculating the number of times passing through the zero point. Yang et al. [22] extracted amplitude feature, amplitude difference feature, triaxial cross-correlation feature, and energy ratio from data in acquisition time window and used OCSVM to distinguish normal data and abnormal data. Siddanahalli Ninge Gowda et al. [23] used acceleration sensor, gyroscope, and magnetic field sensor to collect data, identify the position of acquisition equipment by setting threshold value, and then carry out step counting. Combined with magnetic field sensor to filter the noise generated by random movement, the approach only uses a single sensor for step calculation after identifying the location.

Generally, the step counting process is mainly divided into two stages: filtering and then step counting. The two-stage processing method further improves the ability of the algorithm to deal with the noise. However, the accuracy of noise filtering is required to be high. When the noise filtering is not accurate, it is easy to cause inaccurate step counting. In addition, the updating problem is not taken into consideration in the above approaches.

Because of the shortcomings of the current approaches, this study proposes an intelligent optimization strategy based on deep reinforcement learning. The effective features are extracted by deep learning, and the self-learning ability of reinforcement learning is combined to deal with the noise problem and parameter continuous updating problem. Finally, accurate step counting is realized in the presence of noise, and the parameters are continuously optimized.

3. The Proposed Framework

In this section, the problem is transformed into a series of decision optimization, and deep reinforcement learning framework is introduced to solve the problem (Section 3.1). The procedure of the proposed framework is illustrated (Section 3.2). Then, the third section mainly introduces how to optimize the parameters in the proposed framework (Section 3.3). Finally, Section 3.4 shows the approach details.

3.1. The Deep Reinforcement Learning Framework. Deep learning is a kind of technology with powerful function approximation ability and representation ability, which are mainly to combine multilayer simple but nonlinear modules. In these modules, each layer processes the output of the previous module and continuously transforms the low-level data into a higher-level data representation. Deep reinforcement learning has been applied to various fields of life and achieved excellent results. For example, the generation technology of confrontation network based on deep convolution in Ref. [24] is used to reconstruct the missing information between tomb mural blocks; reinforcement learning is a method for learning optimal sequential decision. The correct decision-making action cannot be known at any time, only a good or bad feedback can be obtained, and the agent needs to learn the feedback. Therefore, agents need to interact with the environment constantly. By making decisions on the input of each moment, the corresponding reward signal is obtained. With the help of the reward signal, the corresponding decision-making mechanism is learned, and the optimal decision-making is finally realized. It is worth noting that the decision at the current moment may have an impact on the subsequent results.

Deep reinforcement learning, which combines the advantages of deep learning and reinforcement learning, can coordinate the learning of original data.

Figure 1 is the framework of deep reinforcement learning. It includes environment (E), agent (A), and sum unit (S). Environment (E) mainly provides input for agent (A), evaluates the action of the agent, and receives the result of sum unit (S). Moreover, the step counting result is evaluated. The evaluation result is returned to the agent (A) as the final reward information r to guide the agent's strategy training. Agent (A) makes a decision and selects the corresponding action who is transported to the environment (E) and sum unit (S); the sum unit (S) accumulates the actions from the agent to obtain the total number of steps.

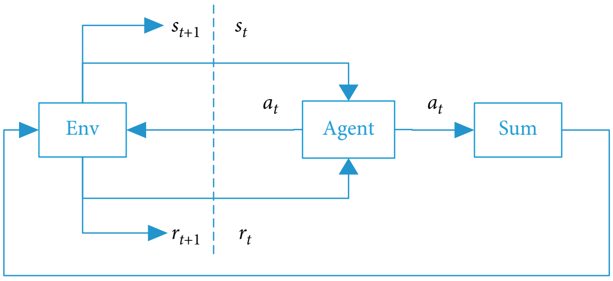

Figure 1: Framework of deep reinforcement learning.

In application, the environment (E) corresponds to the user and the user's data acquisition device, the agent (A) corresponds to the algorithm used in step counting, and sum unit (S) corresponds to the step number accumulation and storage part. The reward information may come from the user's manual input or other input, such as the built-in simple step counting function.

3.2. The Procedure of the Proposed Framework. The main procedure of step counting framework proposed here is as follows: first, the sensor data are collected; then, the agent obtains the data from the environment E and a reward was obtained; and at last, the sum unit accumulates steps.

In the step counting procedure, the agent makes decisions according to the observation obtained from the current environment. We transform the whole step counting procedure into decision-making procedure and establish reinforcement learning model for it. Reinforcement learning mainly includes five parts:

(a) State set: in step counting, the state set contains the permutation and combination of all sensor values in a period. For the state of a certain period, where m is the number of sensor values contained in a certain sampling frequency and a certain sampling time, Assuming that the sampling frequency is c Hz and the duration is T, then $m = c \times T$.

(b) Action set: for each, the agent uses the strategy network to make decisions to obtain the corresponding probability of each action, and select the next action, where 0 indicates that the current state does not need step counting, and 1 indicates the opposite.

(c) Transition probability: in the step counting scene, after the agent selects the action for the current observation, the environment is directly transferred to the next period to obtain the new observation, which is not related to the action in environment, and the transition probability is 1.

(d) Discount factor: when the data have completed the decision in a certain period, and the total number of steps obtained by the sum unit is returned to the environment, the current agent obtains an effective reward. Therefore, the discount factor is more concerned about the reward that can be obtained at the end of step counting, which is also the evaluation of the final step counting result.

(e) Reward R: the sum unit sums the actions selected by the agent in each observation to obtain the final total number of steps. The environment evaluates. Only when the environment obtains the final total steps, the effective reward can be returned. Otherwise, the reward is zero. Specifically, the closer the total number of steps is to the real steps, the higher the reward will be, otherwise the lower the reward will be.

Furthermore, there are two aspects that need to be explained. First, although the transition between the former and the latter states is certain, the decision of different states will affect the result; that is, the actions corresponding to each state will affect the total number of steps; in addition, in this study, the contribution of the actions selected by each state to the final reward is consistent. Second, different users, such as young people, middle-aged people, and the elderly, may not have the same pace frequency when walking; at the same time, the data will be different under different use scenarios, such as running and going upstairs and downstairs. How to extract effective features from data is the first problem to be solved. Therefore, to complete the mapping function of the strategy network, this study uses convolutional neural network (*CNN*) to complete the end-to-end processing and uses the powerful representation ability of neural network to extract useful features from observations.

3.3. Parameter Optimization. In deep reinforcement learning, the parameters in the strategy network are trained by maximizing the expectation of cumulative reward. In this study, the setting of reward function is connected with the accuracy of step counting. The more accurate the step counting is, the greater the reward will be, and the corresponding cumulative reward will be larger. When training the network parameters of the strategy, the existing optimization strategies are solved by deep reinforcement learning to achieve accurate step counting.

To solve the objective function, the optimization problem is solved quickly, which returns the reward value according to the current corresponding strategy. The objective function is as follows:

$$\nabla v_{\pi_\theta}(s) = \nabla \sum_{a} [\pi(a \mid s) q_\pi(s,a)] = \sum_{x\in S}\sum_{k=0}^{\infty} P(s \longrightarrow x, k, \pi)[\nabla \pi(a \mid x) q_\pi(x,a)], \forall s \in S. \quad (1)$$

In equation (1), the Bellman equation is mainly used for expansion, which represents the probability of transition from state to state through steps in strategy. Set $d^\pi(s) = \sum_{k=0}^{\infty} P(s_0 \longrightarrow s, k, \pi)$, and equation (1) can be expressed as follows:

$$\nabla J(\theta) = \nabla v_{\pi_\theta}(s_0) = \sum_{s} d^\pi(s) \sum_{a} [\nabla \pi(a \mid s) q_\pi(s,a)]. \quad (2)$$

Equation (2) is independent of the current state distribution, by replacing $q_\pi(s,a)$,

$$\begin{aligned}\nabla J(\theta) &\approx E_\pi\left[\sum_{a} \pi(a \mid s_t, \theta) q_\pi(s_t, a) \frac{\nabla \pi(a \mid s_t, \theta)}{\pi(a \mid s_t, \theta)}\right]\\ &\approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} \frac{\nabla \pi\left(a_t^i \mid s_t^i\right)}{\pi\left(a_t^i \mid s_t^i\right)} G_t^i,\end{aligned} \quad (3)$$

where $q_\pi(s_t, a_t) = E_\pi[G_t \mid s_t, a_t]$, $G_t^i = \sum_{k=t}^{T} r_k^i$, and N is the times. Gradient descent strategy can be used to solve this optimization function, and the loss function is as follows:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} ln\left(\pi\left(a_t^i \mid s_t^i\right)\right) G_t^i. \quad (4)$$

3.4. The Deep Reinforcement Learning Algorithm. The proposed algorithm is shown in algorithm 1.

In algorithm 1, the first line to the sixth line is the interaction between the policy network and the environment. In the seventh and eighth lines, the data processing of the current period is completed, and the sum unit returns the total step count to the environment. The environment inputs the total number of steps and the real total number y into the reward function to obtain the total feedback of the current round. Lines 9 to 13 update strategy network parameters. The 10th line calculates the cumulative reward G_t of the current time t. Lines 11 and 12 use the cumulative reward G_t and update the gradient of the strategy network.

4. Experimental Analysis

In this section, we conduct experiments on real data set to validate the effectiveness of our approach.

```
Random initialize the parameter of strategy network, Set learning rate parameter, total rounds
for $i = 1, \ldots, N$. do
    Get initialization status $s_0$
    for $t = 1, \ldots, T$. do
        Get $a_t$ according to $\pi(a_t|s_t)$.
        Accumulate the reward according to $a_i$
    end for
    Get the step sum unit $y'$
    Get reward according to $y'$
    for $t = 1, \ldots, T$. do
        $G_t = \sum_{k=t}^{T} r_k$.
        According to formula (4), optimizing parameters
        Updating parameters: $\theta \leftarrow \theta - \alpha \cdot \nabla L(\theta)$.
    end for
end for
```

Algorithm 1: The proposed approach.

4.1. Data Set. The data set used in this section is provided by Cambridge University [25], which not only contains the walking data, but also collects the noise data, and the duration of the two is roughly the same. The main features of the data set are shown in Table 1.

The data set in Table 1 is collected by Cambridge University according to the walking data collected by the test object holding the mobile phone in various states, including each participant walking the same distance in various states, such as holding the mobile phone in his hand, in his pocket, in his backpack, or in handbag. At the same time, the route is divided into three stages. The walking states are pleased, acceleration, and deceleration.

The data acquisition frequency was 100 Hz. There were 27 young participants, including 18 men and 9 women, and a total of 130 walking routes. The height of all participants was in the normal range. The data came from sensors at six positions, including hand, typing on hand, front and rear pockets of trousers, handbags, and backpacks.

The collected data samples record the start time and end time of walking. The data not in this time period are regarded as noise data. The data set comprehensively collects the current young people's use of mobile phones in various scenarios.

4.2. Data Preprocessing. In the process of collecting original data, the higher the acquisition frequency, the more sensitive the sensor is. Therefore, all kinds of tiny noises can be captured, so it is necessary to preprocess the original data. On the other hand, deep learning is used to learn data representation in deep reinforcement learning. To make the network better for feature learning, it is necessary to preprocess the data, such as unifying the dimensions of the data, to facilitate the matrix operation in the neural network. In this study, the data preprocessing operations are as follows:

(1) Calculate the resultant acceleration and remove the gravity: in step counting, it is considered to calculate the combined acceleration value for the three components. Considering that the gravity value is always a fixed value in a certain environment, and to better reflect the positive and negative characteristics of the data, the gravity acceleration value of 9.80665 is removed, and the formula is $a_{norm} = acc - 9.80665$ 9.

(2) Down sampling: in deep learning, the model parameters are not only related to the network structure, but also related to the input data. Considering the operation efficiency, and to remove the small noise, the data obtained in this study are sampled without overlapping. The sampling formula is $acc_i' = \sum_{k=0}^{l} acc_{i+k}$. acc_i' is the data after down sampling.

In Figure 2, the y-coordinate is the range of the value after subtracting the gravitational acceleration 9.80665; therefore, the value of the y-coordinate in Figure 2 floats up and down 0. The x-coordinate is the sampling point, and the size of the lower sampling window is 4. Figure 2 shows the comparison of data before and after subsampling, and the red part indicates that the original small noise points disappear after down sampling. As can be seen, the sampled data can not only remove small noise, but also reduce the amount of data and reduce the input data dimension of the model.

(3) Data segmentation and filling: the size and length of input data need to be set in advance when using convolutional neural network, so the dimension of input data needs to be fixed. In application, the length of the data collected by sensors increases with the operation of the acquisition equipment. It is necessary to segment the data to fit the preset size of the neural network. To unify the segmented data of different lengths, it is necessary to fill and crop the segmented data. In the filling process, the destruction of the original data should be reduced as much as possible. The left and right values should be used to copy and fill, so that the filled data have a uniform length without destroying the original data waveform. In the cutting process, the original data

Table 1: Characteristic description of experimental data set.

Data set	Cambridge University data set
Acquisition frequency	100 Hz
Number of people collected	27
Sex ratio	2 (M) : 1 (F)
Age range	15–29
Height range	150–189 (cm)
Acquisition location	Hands, front pant pockets, back pant pockets, hand typing, handbags, and backpacks
Change in walking speed	Random walking ⟶ accelerating walking ⟶ decelerating walking
Data marker	Total number of steps and the start and end times of walking
Ratio of noise duration	47%

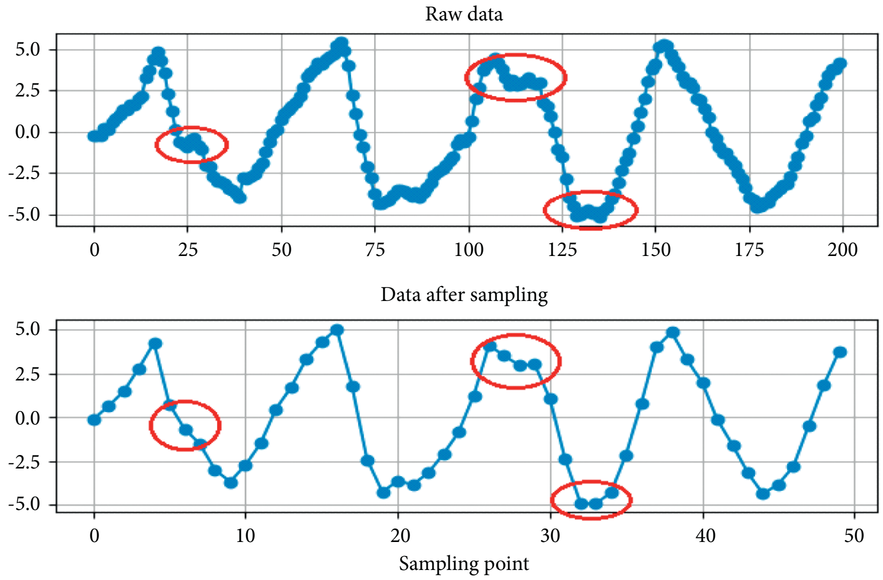

Figure 2: Data before and after down sampling.

waveform should not be damaged as much as possible, so the data on the left and right sides can be trimmed at the same time.

In Figures 3 and 4, when the mobile phone is placed on the hand, the acceleration collected by the mobile phone sensor changes. In Figure 3, the acceleration changes increase first and then decrease, and the peaks and troughs appear alternately. Using that, the peaks and troughs only appear once in each step, and taking the peaks and troughs as the dividing point between the current step and the subsequent step, without damaging the front and rear waveforms, the data are quickly segmented.

After the data are segmented, the length of the segmented data is inconsistent due to different walking speeds, different positions of the acquisition equipment, and walking on different roads. Taking Figure 3 as an example, after the data are divided, the data length appears 11, 13, and 15. To unify the segmented data with different lengths, it is necessary to fill and cut the segmented data. The comparison before and after segmentation and fixed length of 21 is shown in Figure 4.

4.3. Comparisons. To show the performance improvement of the proposed approach, we compare the proposed approach with several common approaches or commercial products: peak detection, autocorrelation coefficient, frequency-based method using acceleration, Spring Run, and Ledongli. In addition, we choose to use accuracy that is defined as follows as metric to evaluate the results:

$$\text{accuracy} = 1 - \frac{1}{M}\sum_{i=1}^{M}\left|y_i' - y_i\right|. \tag{5}$$

where y_i' is estimated number of steps andy_i is the actual number of steps. In this study, firstly, the training data and the test data are randomly divided into 1 : 1, several groups of experiments are carried out, and the average value is taken as the result of the proposed framework.

4.4. Experimental Results. This experiment is based on the framework of deep reinforcement learning. In the strategy network, parameter values have an impact on the results of the experiment. Firstly, the acceleration data are processed and used as the input of the model. For convenience, the convolution kernel length of convolution layer is set to 21, and the convolution kernel width is 1 because the resultant

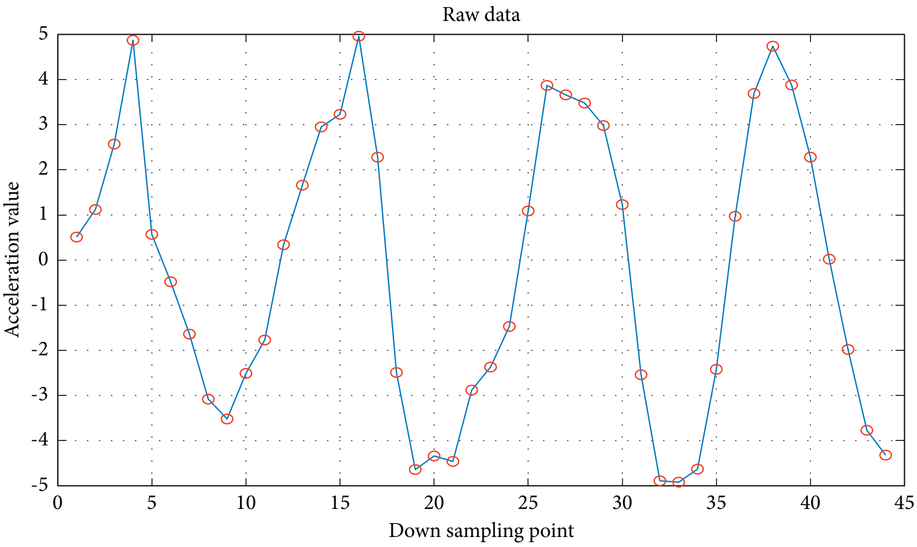

FIGURE 3: Data before segmentation and filling.

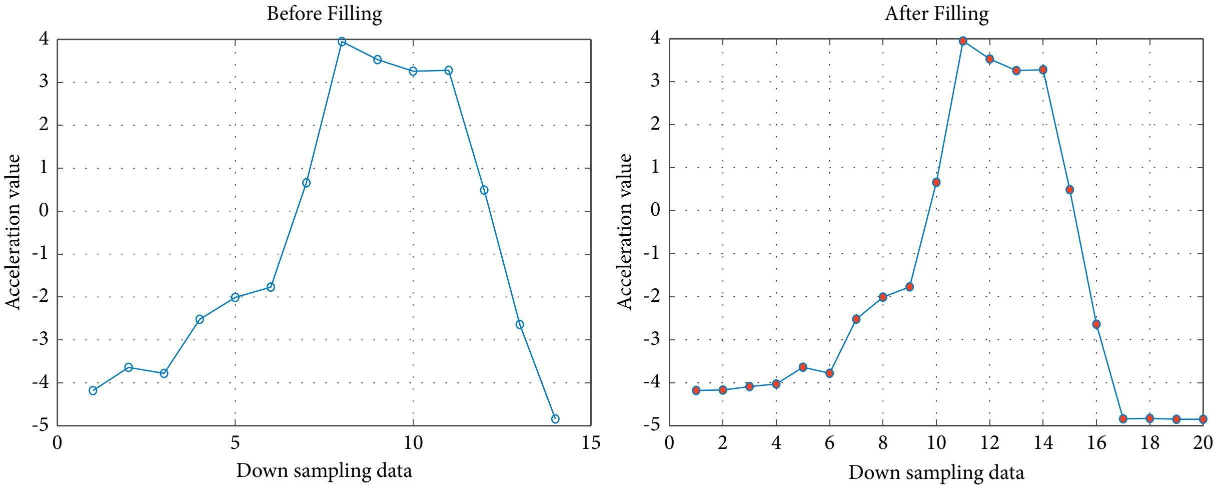

FIGURE 4: Data after segmentation and filling.

acceleration data are one-dimensional. In this experiment, in the activation function, except for the output layer, ReLU is used, and dropout is used after splicing. To find a reward function suitable for the model, experiments are carried out on a variety of reward functions:

$$R_{base}(x, y) = \frac{1}{\sqrt{2\pi}y/2}\exp\left(-\frac{(x-y)^2}{2(y/2)^2}\right). \tag{6}$$

In this study, the influence of learning rate on the step counting error is shown in Figure 5 without adding three layers of fully connected layer. In Figure 5, the x-coordinate is the training rate and the y-coordinate is the error.

From the overall trend, the step counting errors of the four models first decrease and then increase with the increase in learning rate and reach the trough between the learning rates of 0.0001 and 0.01.

At this time, the effects of the two methods are similar, especially when the learning rate is between 0.0005 and 0.01. The step counting error almost does not change with the change in learning rate, showing good stability. It can be seen that when the number of convolution kernels is 30, the effect is better than the model with 10 and 20 convolution kernels. At this time, the performance of the model will not be greatly improved by increasing the number of convolution kernels. Therefore, the model with 30 convolution kernels is considered as the basic model for subsequent experiments.

The average accuracies of different approaches and products are shown in Figure 6. Deep reinforcement learning (DRL) is the framework proposed in this study; AC [6] represents autocorrelation coefficient algorithm; PD [10] represents peak detection algorithm; and FA [15] represents frequency-based method using acceleration. It can be seen that the accuracy of the proposed approach (DRL) is close to the optimal and its average accuracy is as high as 97.28%, which is higher than both of the comparing traditional approaches and commercial products by at least 5.01%. In complex cases, the accuracy of Spring Run and PD-based

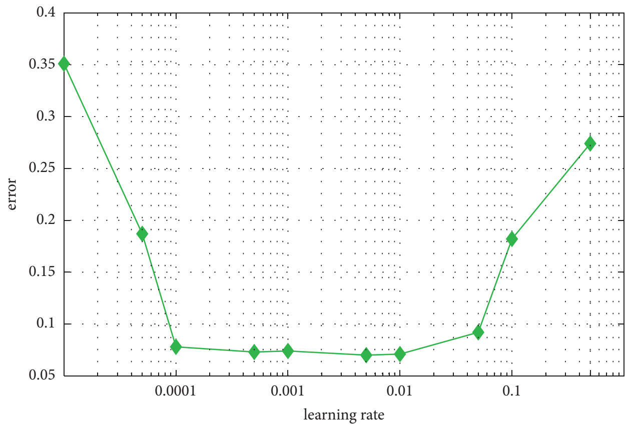

Figure 5: Influence of different learning rates.

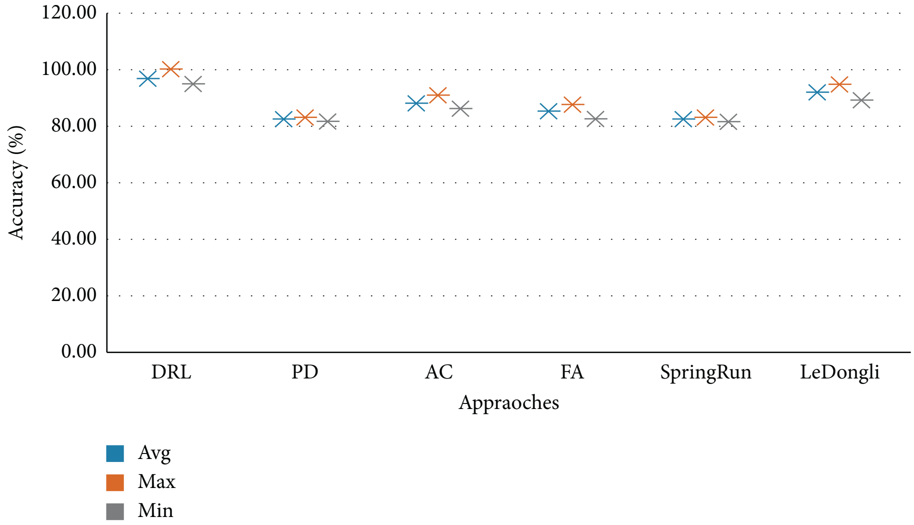

Figure 6: Comparison of step counting results of different approaches.

algorithms is relatively low, and the accuracy is only about 80%. In addition, the blue color represents the average accuracy, and red and grey represent the best and worst accuracy in complex situations.

5. Conclusions and Future Work

In the step counting process of traditional algorithm, a two-stage processing method is used; that is, the noise is filtered first, and then, the steps are counted. To alleviate the problem of inaccurate step counting caused by inaccurate noise filtering and continuously optimize the model parameters, a step counting framework based on deep reinforcement learning is proposed. Our main contributions are as follows:

(1) Here, the step counting is transformed into a serialization decision optimization problem, and an intelligent optimization strategy based on deep reinforcement learning is designed. It is an end-to-end direct solution.

(2) Down sampling, segmentation, and filling techniques are used to preprocess data, which reduce the dimension of the input data. Figures 2 and 3 are comparison results before and after data down sampling, segmentation, and filling. In addition, the interaction between users is utilized to update parameters by user feedback while identifying noise.

(3) Experiments show that the proposed framework could obtain more effective and accurate results. Figure 4 shows the trend of step counting error

with the change in learning rate, so that it can be determined in the deep reinforcement learning framework to improve the step counting accuracy. Figure 5 shows the accuracy comparison between the proposed model and the existing pedometers or step counting algorithms on the same data set, and it shows that as an end-to-end direct solution, deep reinforcement learning can effectively improve the accuracy and reduce the error.

However, the update of strategies depends on the current payoff, which may lead to slow learning and large variance. In value-based reinforcement learning, value function and state action function are used to estimate the expected return value, which will cause information loss. In the future work, we can model the value distribution, reduce the instability in the value function estimation, reduce the decision-making risk, and provide more choices for the predicted values. The future work may consider introducing parallel computing methods, more feedback information, more features, etc.

Data Availability

The data set used in this section is provided by Cambridge University, please visit https://dl.acm.org/doi/10.1145/2493432.2493449.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Authors' Contributions

Many thanks are expressed to Xiaodong Zhang for his kind help during the preparation of the manuscript and to Pengfei Chen for assistance with the experiments.

Acknowledgments

This work was supported by the grants from the National Key Research and Development Program of China (No. 2019YFB1404602).

References

[1] S. Tolonen, H. Sievänen, M. Hirvensalo et al., "Higher step count is associated with greater bone mass and strength in women but not in men," *Archives of Osteoporosis*, vol. 13, no. 1, p. 20, 2018.

[2] D. Haritha, J. Mounika, and G. Singh, "An adaptive handheld inertial pedestrian navigation system," *Journal of Information Storage and Processing Systems*, vol. 20, no. 2, pp. 309–323, 2021.

[3] E. James and A. Rahimanl, "Prevalence of caries and treatment needs among 15-17-year-old students in Calicut district: a cross-sectional study," *International Journal of Medical Sciences*, vol. 6, no. 6, pp. 92–98, 2020.

[4] H. Li, N. Kumar, and R. Chen, "Deep reinforcement learning," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Alberta, Canada, April 2018.

[5] H. S. Zhang, L. Li, and X. Liu, "Development and test of manhole cover monitoring device using LoRa and accelerometer," *IEEE Transactions on Instrumentation and Measurement*, vol. 99, p. 1, 2020.

[6] W. Y. Hu, J. L. Lu, and S. Jianget al, "WiBEST: A hybrid personal indoor positioning system," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, vol. 7, Shanghai, China, April 2013.

[7] B. Huang, G. Qi, and X. Yang, "Exploiting cyclic features of walking for pedestrian dead reckoning with unconstrained smart phones," in *Proceedings of the ACM International Joint Conference*, pp. 374–385, Heidelberg, Germany, September 2016.

[8] X. L. Chen, L. Cao, and Z. X. Xu, "Ensemble network architecture for deep reinforcement learning," *Discrete Dynamics in Nature and Society*, vol. 2018, Article ID 2129393, 6 pages, 2018.

[9] S. Jayalath and N. Abhayasinghe, "A gyroscopic data based pedometer algorithm," in *Proceedings of the International Conference on Computer Science and Education*, pp. 551–555, Colombo, Srilanka, April 2013.

[10] Y. Lu and S. Velipasalar, "Autonomous footstep counting and traveled distance calculation by mobile devices incorporating camera and accelerometer data," *IEEE Sensors Journal*, vol. 17, no. 21, pp. 7157–7166, 2017.

[11] V. Genovese, A. Mannini, and A. M. Sabatini, "A smartwatch step counter for slow and intermittent ambulation," *IEEE Access*, vol. 5, pp. 13028–13037, 2017.

[12] C. Luo, A. Visuri, S. Klakegg et al., "Energy-efficient prediction of smartphone unlocking," *Personal and Ubiquitous Computing*, vol. 23, no. 1, pp. 159–177, 2019.

[13] C. Liu and H. Yang, "Research on adaptive step detection algorithm," *Piezoelectric and acoustooptic*, vol. 37, no. 2, pp. 258–261, 2015.

[14] G. H. Wang, J. Z. Liang, and J. Chen, "Acceleration differential finite state machine step counting algorithm," *Computer science and exploration*, vol. 10, no. 8, pp. 1133–1142, 2016.

[15] Z. Tang, Y. Guo, and X. Chen, "Self-adaptive step counting on smartphones under unrestricted stepping modes," in *Proceedings of the 40th IEEE Annual Computer Software and Applications Conference*, pp. 788–797, Atlanta, GA, USA, June 2016.

[16] A. Rai, K. Chintalapudi, and V. N. Padmanabhan, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, pp. 209–304, Istanbul, Turkey, August 2012.

[17] F. Xiao, "Design of step detection algorithm based on acceleration threshold detection," *Software Engineering and Applications*, vol. 9, no. 1, pp. 93–110, 2020.

[18] J. W. Kim, H. J. Jang, and D. H. Hwang, "Stride and step length obtained with inertial measurement units during maximal sprint acceleration," *Sports*, vol. 7, no. 9, p. 202, 2019.

[19] X. Liu, N. Li, and G. Xu, "A novel robust step detection algorithm for foot-mounted IMU," *IEEE Sensors Journal*, vol. 99, p. 1, 2020.

[20] R. Delgado-Gonzalo, P. Celka, and P. Renevey, "Physical activity profiling: activity-specific step counting and energy expenditure models using 3D wrist acceleration," in *Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 8091–8094, Milan, Italy, August 2015.

[21] L. Becchetti, E. Cruciani, F. Pasquale, and S. Rizzo, "Step-by-step community detection in volume-regular graphs," *Theoretical Computer Science*, vol. 847, no. 2, pp. 49–67, 2020.

[22] C. Yang, J. H. Yan, and C. J. Guo, "Fault detection of fault-tolerant integrated navigation based on one-class SVM," *Computer Simulation*, vol. 32, 2019.

[23] A. K. Siddanahalli Ninge Gowda, S. R. Babu, and D. C. Sekaran, "UMOISP: usage mode and orientation invariant smartphone pedometer," *IEEE Sensors Journal*, vol. 17, no. 3, pp. 869–881, 2017.

[24] M. Wu, A. Payshanbiev, and Q. Zhao, "Nonlinear optimization generating the tomb mural blocks by GANS," *Applied Mathematics and Nonlinear Sciences*, vol. 6, no. 1, pp. 43–56, 2020.

[25] A. Brajdic and R. K. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 225–234, Zurich, Switzerland, September 2013.