IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 17 November 2022, accepted 9 December 2022, date of publication 14 December 2022, date of current version 21 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3229044

RESEARCH ARTICLE

An Elderly Fall Detection Method Based on Federated Learning and Extreme Learning Machine (Fed-ELM)

ZHIGANG YU[1,*], JIAHUI LIU[2,*], MINGCHUAN YANG[1], YANMIN CHENG[1], JIE HU[1], AND XINCHI LI[1]
[1]Institute of Big Data and Artificial Intelligence, China Telecom Research Institute, Beijing 102209, China
[2]Institute of Automation, Shenyang Aerospace University, Shenyang 110136, China

Corresponding authors: Zhigang Yu (yuzhg@chinatelecom.cn) and Jiahui Liu (3060212475@qq.com)

This work was supported by the Key Research Projects of China Telecom Research Institute ''Research on the Technical Architecture and Key Technologies of Data Element Value'' under Grant P-2022-03.

*Zhigang Yu and Jiahui Liu contributed equally to this work.

ABSTRACT The lack of fall data for the elderly is a challenging problem in the fall detection community. To date, the fall and activities of daily life simulated by young people have been used in most studies to train and test fall detection algorithms. However, there are differences in movement patterns between young and elderly individuals due to bone aging, which leads to the degradation of the algorithm performance in the elderly population. To solve the above issue, this paper proposes a fall detection algorithm combining Federated Learning and Extreme Learning Machine (Fed-ELM). First, the online extreme learning machine can use a small amount of misclassified user data to update the parameters so that its performance is improved for individual users. Then, Federated Learning is applied to share data information among different users without involving user privacy. In this way, the generalizability of the fall detection algorithm is improved. The performance of the proposed algorithm in different age groups is analyzed by experiments. For young people, the accuracy, sensitivity and specificity reach 96.96%, 94.50% and 99.29%, respectively, and the accuracy on each individual is more than 94%. For elderly individuals, the accuracy, sensitivity and specificity reach 99.07%, 96.00% and 98.33%, respectively, and the accuracy of each individual is more than 96%.

INDEX TERMS Fall detection, federated learning, extreme learning machine, elderly fall.

I. INTRODUCTION

Globally, falls are the second leading cause of accidental injury after traffic accidents, and more than 30% of the elderly experience at least once fall each year. Among the fallen elderly, 90% cause hip bone fracture, and 60% cause head injury [1]. In addition, nearly half of the elderly lie on the ground for a long time (long-lie) after a fall, which causes injuries, such as pressure sores, dehydration, hypothermia, pneumonia, and even death [2]. Therefore, a fall detection system that can inform nursing for the fallen elderly in a timely manner is necessary. The fall detection system can shorten the time for medical assistance arrival and minimize the injury caused by the fall.

At present, fall detection methods can be divided into nonwearable-based fall detection methods [3], [4], [5] and wearable-based fall detection methods [6], [7], [8], [9]. Nonwearable-based fall detection methods include visual fall detection methods [10], [11] and environmental fall detection methods [12]. The visual fall detection method uses a camera to identify falls and Activities of Daily Life (ADL, which means types of activity such as walking and running). The fall detection method based on the environment places sensors, such as pressductors and infrared sensors in the activity area of the users to detect falls. However, it only works in a limited space. When the elderly leave the detection

The associate editor coordinating the review of this manuscript and approving it for publication was Jad Nasreddine.

area, the system cannot monitor their activity. The wearable fall detection method places sensors, such as accelerometers and/or gyroscopes in a certain part of the human body to monitor the activity changes of users. It has the advantages of unlimited space, convenient use, small size and low price. Therefore, the wearable fall detection method is the focus of this paper.

In recent years, researchers have mainly focused on the performance of wearable fall detection algorithms [21], [24]. Nevertheless, there are some other factors to consider in real world scenarios. For example, the elderly fall data sample is small, and the collection cost is high. Most of the current studies use youth data instead of elderly data, but the inconsistent data distribution weakens the performance of the fall detection algorithm. Therefore, sharing the activity information of the elderly under the condition of privacy protection is also worth considering. In addition, the difference in consumers' motion posture also leads to the degradation of accuracy. As a result, it is necessary to design a fall detection algorithm that can dynamically adjust the parameters to adapt to various individuals.

To address the problems raised above, an elderly fall detection method based on Federated Learning and Extreme Learning Machine (Fed-ELM) is proposed in this paper. First, the Extreme Learning Machine (ELM) is trained with the young data to obtain the initial fall detection model. Then, the parameters of ELM are updated online by the sparse substantial data of the consumer (OS-ELM), which can make the fall detection algorithm more applicable to the consumer. When OS-ELM runs for a period of time, the algorithm parameters of each consumer are uploaded to the cloud platform for federated aggregation. Finally, the parameters obtained by Federated Learning (FL) are delivered to the consumer to improve the robustness of the fall detection algorithm. In this way, the proposed fall detection algorithm can share information of different individuals while protecting privacy and update the parameters according to each individual's characteristics. As a result, the performance of the proposed fall detection algorithm is improved.

This paper is organized as follows: The fundamental theory involved in this paper is introduced in section II. The working principle of the proposed Fed-ELM is presented in section III. The experimental process and result analysis of this paper are displayed in section IV. Finally, this paper is concluded in section V.

II. FUNDAMENTAL THEORY

A. FALL DETECTION METHOD

Algorithms for wearable fall detection methods can be divided into two categories: 1) threshold-based fall detection methods [13], [14], [15], [16], [17], [18], [19], [20] and 2) machine learning-based fall detection methods [21], [22], [23], [24].

Threshold-based fall detection method: In early fall detection research, Threshold Based Method (TBM) with small computation was the most frequently applied. In earlier studies, Bourke et al [13] used accelerometers and gyroscopes to collect information on human movement and compared the set thresholds with the extracted acceleration- and angular velocity-related features to identify whether falls occur. Finally, the sensitivity and specificity reached 100% in 240 sets of simulated falls and 480 sets of ADLs. In recent years, Wang et al [14], [15], [16], [17], [18], [19], [20] used the TBM algorithm to detect falls to reduce the power consumption of wearable devices. In their latest study, they used two 3-axis accelerometers and a barometer to detect falls, but the sensitivity only reached 91%.

The fall detection method based on machine learning: Compared with the TBM fall detection algorithm, the fall detection method based on machine learning has better performance. In the latest research, Nicolas Zurbuchen et al [21] used the K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Gradient Boosting (GB) to distinguish between falls and ADLs. The experimental results showed that all five algorithms perform well except the SVM algorithms. The sensitivity and specificity of the DT and RF algorithms were close to 99%. Eduardo et al [26] used Convolutional Neural Network (CNN) to detect falls and analyzed its performance at different sampling frequencies and various locations. The results showed that the performance is satisfactory. Additionally, when the sampling frequency is 20 Hz, the global quality metric exceeds 90% on most public datasets. Nho et al [23] proposed the Generative Adversarial Network (UI-GAN) for anomaly detection and fall recognition. They used the initial information to generate approximate nonfall samples and compared them with the samples that needed to be identified. Ultimately, the method is more suitable for different individuals and achieves an accuracy, sensitivity and specificity of 96.94%, 98.23%, and 95.87%, respectively. Before the execution of UI-GAN, 36 features extracted from the original signal are processed as 36 pixels of the image. Then, the user's initial information is used to generate approximate nonfall samples. The generated nonfall samples are compared with the samples that need to be identified. When the distance between them is further than the set value, it is determined as abnormal data (fall). It can be concluded that compared with the threshold-based fall detection algorithm, the fall detection algorithm based on machine learning performs better.

B. TRADITIONAL ELM

Extreme Learning Machine [28] is a single hidden layer neural network with forward propagation. Compared with the traditional feedforward neural network, ELM has the advantages of fewer training parameters, faster learning speed and stronger generalizability. The main training process of the algorithm can be divided into two parts: 1) parameter random initialization and 2) linear parameter solving.

In the parameter random initialization stage, the hidden layer parameters are randomly initialized, and the activation function is determined. The activation function is a nonlinear mapping that maps the input data to the ELM feature space. Specifically, parameter initialization randomly generates the weight (w) and bias (b) of the hidden layer nodes. In contrast to many existing machine learning methods, the feature mapping stage can be arbitrary nonlinear piecewise continuous functions. Furthermore, because the node parameters of the hidden layer are randomly generated, it performs better in efficiency. Subsequently, N training samples $X = (x_1, x_2 \ldots\ldots, x_N)$, and corresponding labels $t = (t_1, t_2 \ldots\ldots, t_m)$ and determinate hidden layer node parameters can be used to calculate the output of hidden layer H (Formula 1).

$$H(w, b, X) = g(w^* X + b) \tag{1}$$

In the linear parameter solving stage, the output layer weight β needs to be solved first. To ensure the performance of the ELM on training sample X, the error $e = [\beta H - T]^2$ between the network output and the sample label must be minimized. In the traditional gradient descent, all parameters can be adjusted by continuous iteration, but H is fixed here, so the training process only needs to solve the output weight in the equation $\beta H = T \beta H = T \beta H = T \beta H = T$. Finally, the output weight can be calculated by Formula 2:

$$\hat{\beta} = H^{-1} T \tag{2}$$

where H^{-1} is the Moore-Penrose generalized inverse of the matrix H. It can be proven that the norm of the solution β obtained is minimal and unique.

C. FEDERATED LEARNING

Federated Learning has been applied to many fields [25], [26], [27], [29], and the goal is to solve the contradiction between data islanding and data privacy protection. By establishing a data federation, efficient machine learning can be carried out among multicenters or multicomputing nodes under the premise of ensuring data privacy security and legal compliance. In short, FL enables comodeling based on multicenter data while protecting data privacy and security and driving the continuous development of AI technology.

The main components of FL include the local database, local computer, central server, etc. The central server can be set up in the cloud independent of the centers participating in FL, or set up in the local center participating in FL. Since the information shared between each center and the server is only the model's parameters, private data of wearable device users are not exchanged.

FedAvg is evolved from the stochastic gradient descent algorithm, FedSGD [30]. First, the weights of the neural network model are initialized on a trusted third party (central server); then, the weights are assigned to the participants to train the local model, and the process stops after a certain number of iterations. For each global polling, participants of proportion C are selected for iteration, the total sample number of all participants is set as n, the number of data samples owned by the m-th participant is n_m, and the objective function to be optimized is $\min \min_{k \in R^d} f(k)$.

$$f(k) \stackrel{\text{def}}{=} \frac{1}{n} \sum\nolimits_{i=1}^{n} f_i(k) \tag{3}$$

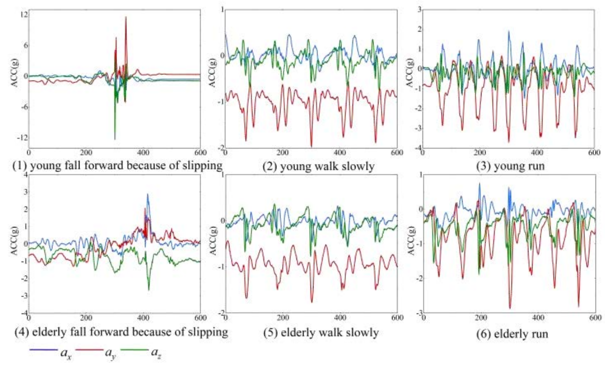


FIGURE 1. Typical fall and ADL of young and elderly people.

$f_i(k) = l(x_i, y_i; k)$ is the predicted loss of sample (x_i, y_i) for model parameter w. For the m-th participants:

$$F_m(k) = \frac{1}{n_m} \sum\nolimits_{i \in P_m} f_i(k) \tag{4}$$

Then, the overall loss function of the federation model is:

$$f(k) = \sum\nolimits_{m=1}^{M} \frac{n_m}{n} F_m(k) \tag{5}$$

The gradient of m participants is $g_m = \nabla F_m(k_t)$, and the learning rate is a. Then, the new parameter obtained in the t round iteration is:

$$k_{t+1} \leftarrow k_t - \alpha \sum\nolimits_{m=1}^{M} \frac{n_m}{n} g_m \tag{6}$$

The local update for each participant is:

$$k_{t+1}^{m} \leftarrow k_{t}^{m} - \alpha \nabla F_m(k_t) \tag{7}$$

In [30], FedAvg is used to recognize images and natural language. The results show that under the same conditions, compared with Fed-SDG, FedAvg can effectively classify images and natural language. As a result, FedAvg is robust.

III. METHODOLOGY

In the area of fall detection, elderly fall data are scarce and difficult to collect. Most studies use data from young people to train and test fall detection algorithms. However, the gait robustness and activity range of young and elderly individuals are different due to their contrasting movement patterns. As shown in Figure 1, when the same type of fall (falling forward because of a slip) occurs in young and elderly individuals, there are differences in the transformation law and fluctuation amplitude of acceleration. Furthermore, two types of ADLs with different levels of activity intensities are compared. It can be seen that in the case of small activity intensity (walking slowly), the transformation law and fluctuation amplitude of acceleration are similar for both elderly and young individuals. However, when the activity intensity increases, the elderly individual's fluctuation amplitude of acceleration is smaller than that of a younger individual, and the transformation law of the young is more regular. Therefore, the experimental results obtained with young data cannot represent the performance of the proposed method in detecting elderly falls. In Figure 2, Principal Component Analysis (PCA) is used to reduce the ten features extracted in

TABLE 1. Ten fall-related features extracted from raw acceleration data.

Number	Feature Description	Represented Stage
1-3	$SAVA_{x/y/z} = \sum_{i \in DW} \lvert a_{x/y/z}[i] \rvert$	Weightless and impact
4	$MAV_{\min} = \min_{i \in DW} MAV[i]$	Weightless
5	$MAV_{\max} = \max_{i \in DW} MAV[i]$	Impact
6	$AVM_{\min} = \min_{i \in DW} AVM[i]$	Weightless
7	$AVM_{\max} = \max_{i \in DW} AVM[i]$	Impact
8	$AVM_{pp} = AVM_{\max} - AVM_{\min}$	Weightless and impact
9	$AVM_{mean} = \frac{1}{N_{DW}} \sum_{i \in DW} AVM[i]$	Static
10	$\overline{a_y} = \frac{1}{N_{DW}} \sum_{i \in DW} \lvert a_y[i] \rvert$	Static

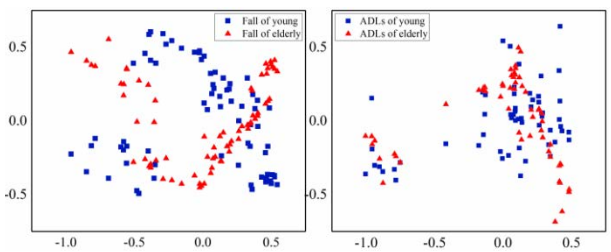


FIGURE 2. Distribution of falls and ADL data in young and elderly people.

Section 3.1 to two dimensions, and scatter distribution maps of young and elderly falls and ADLs are drawn. It can be concluded that there are differences in the distribution of data between young and elderly individuals, especially when falls occur. Thus, a fall detection algorithm for the elderly that can realize user information sharing and apply to different individuals is proposed in this paper.

The proposed fall detection method consists of feature extraction and a fall detection algorithm (Fed-ELM). First, it is necessary to extract the features that can characterize the different stages of falls. To eliminate the undesirable effects of singular samples, the extracted features are normalized. Then, the Fed-ELM can adjust parameters for different individuals and realize information sharing uses the extracted features to distinguish falls from ADLs.

A. FEATURE EXTRACTION

a_x, a_y and a_z are the 3-axis acceleration data obtained after processing the raw data collected by the sensor. In this paper, a_x, a_y and a_z are used to extract two features widely used in the field of fall detection: the 3-axis acceleration magnitude vector (AVM) and the maximum absolute value of the 3-axis acceleration (MAV)[31], [32], [33], [34]. They are also used to further extract new features, and the calculation formulas are as follows:

$$AVM[i] = \sqrt{a_x^2[i] + a_y^2[i] + a_z^2[i]} \tag{8}$$

$$MAV[i] = \max\{|a_x[i]|, |a_y[i]|, |a_z[i]|\} \tag{9}$$

where $a_x[i]$ indicates that it is the i-th acceleration sample along the x-axis ($a_y[i]$ and $a_z[i]$ are similar). In this paper, ten features that can characterize different stages of falls are extracted, as shown in Table 1.

1) $SAVA_{x/y/z}$: The sum of the absolute values of a_x, a_y or a_z can characterize the activity intensity of the subjects along the x-, y-, or z-axes. For the direction of gravity, when the subject is in the weightless stage, the activity intensity in the direction of gravity decreases. When the subject hits the ground, the activity intensity in this direction is much higher than that of ADLs. Therefore, $SAVA_{x/y/z}$ can represent the weightless and impact stages of a fall.

2) MAV_{min} and AVM_{min}: The minimum values of MAV and AVM. While falling, the subject experiences the weightless stage, and the acceleration in the direction of gravity drops rapidly at this stage. Thus, MAV_{min} and AVM_{min} are less than 1 g in the weightless stage. Therefore, these two features can characterize the weightless stage of a fall.

3) MAV_{max} *and* AVM_{max}: The maximum values of MAV and AVM. While falling, the subject experiences the impact stage, and the acceleration in the direction of gravity of the subject rises rapidly at this stage. Thus, MAV_{max} *and* AVM_{max} produce a larger peak during the impact stage. Therefore, these two features can characterize the impact stage of a fall.

4) AVM_{pp}: The value of AVM_{max} minus AVM_{min}. It can represent the dynamic transition of the fall between the weightless stage and the impact stage.

5) AVM_{mean}: The average of the $AVMs$. It can detect the static activities of the subject, such as lying down, sitting, and standing.

6) $\overline{a_y}$:Mean value of the y-axis acceleration. When lying down, the body is parallel to the direction of gravity ($a_y \approx 0\ g$), and the stationary stage of a fall can be identified.

To verify whether the proposed features can effectively distinguish falls from ADLs, the ten extracted features are analyzed. As shown in Figure 3, there are substantial differences in the ten features extracted from falls and ADLs.

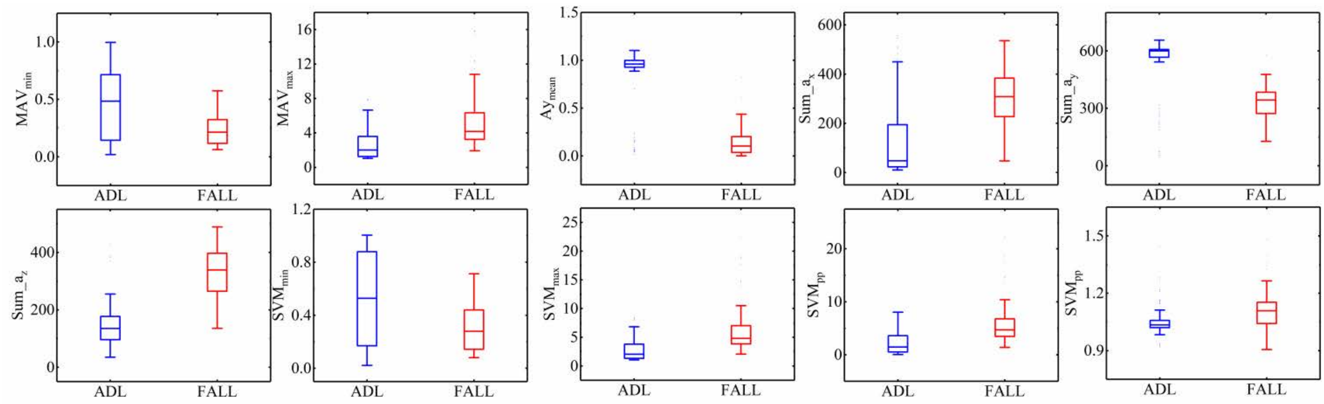


FIGURE 3. Box plots of falls and ADLs for ten characteristics.

Algorithm: The Workflow of Proposed Fed-ELM

Training of the ELM:

1. Initialize weights (w) and biases (b) randomly
2. Calculate the output matrix (H_0) of the hidden layer
3. Calculate the output weight $\beta^{(0)} = P_0 H_0^T T_0$, where $P_0 = (H_0^T H_0)^{-1}$, $T_0 = [t_1, t_2 \ldots t_N]^T$

Client K: Update the OS-ELM with the labeled data:

4. The '*update flag*' of all clients are set to 0
5. If a locally collected sample is labeled, calculate the output matrix (H_i) of the hidden lay
6. Calculate the output weight $\beta^{(i)}$:
7. Return the output weight β
8. Assign the '*update flag*' to client K as 1

Federated Learning:

9. ***for*** each round: **t = 1,2,3......, Do**
10. The clients with '*update flag*' 1 are selected
11. ***for*** the selected client K, **Do**:
$$(w_{t+1}^k, \beta_{t+1}^k, b_{t+1}^k) = Update\ Client(k, w_t, \beta_t, b_t)$$
12. $(w_{t+1}, \beta_{t+1}, b_{t+1}) = \sum_{k=1}^{K} (w_{t+1}^k, \beta_{t+1}^k, b_{t+1}^k)$
13. Return $(w_{t+1}, \beta_{t+1}, b_{t+1})$ to the clients
14. Return to Step 4.

FIGURE 4. Workflow of the Fed-ELM Algorithm.

B. FED-ELM-BASED FALL DETECTION ALGORITHM

The fall detection algorithm designed in this paper consists of two parts: OS-ELM, which can update parameters online according to personal information, and the Federated Learning algorithm, which can share data among different individuals. The execution process of the fall detection algorithm is dispalyed in Figure 4.

First, the ELM algorithm is trained by young data, and then the trained algorithm is arranged in each local client to detect human activity. Next, the trained ELM is executed on the client side. In case a false alarm occurs, the misjudged events are manually labeled, and the OS-ELM algorithm is working (update the parameter, β, of ELM). After the OS-ELM algorithm is executed in the local client for a period of time, the parameter information of the OS-ELM is uploaded to the cloud platform for federated aggregation. Then, the aggregated result is fed back to each client. As a result, the local data are shared, and uploading parameter information avoids the possibility of personal privacy leakage.

For ELM, the dataset $X = [x_1, x_2 \ldots\ldots, x_N]$ and the corresponding label $T = [t_1, t_2 \ldots\ldots, t_N]$ are used for training where N is the number of training samples. In the feature extraction stage, ten features that can characterize different stages of falling are selected, so the input vector $x_i = [x_{i1}, x_{i2} \ldots.., x_{i10}]$, and the data label t_i is 1 or 2 (two classification events, where 1 represents a fall and 2 represents an ADL). In this paper, the number of neurons in the hidden layer of the ELM is set to 80, and the activation function $h(\cdot)$ is the *tanh* function. First, the input weight w_i and bias b_i of the ELM are randomly initialized, and then the initial output matrix H_0 of the hidden layer is calculated, as shown below:

$$H_0 = \begin{bmatrix} h(w_1, b_1, x_1) & \ldots\ldots & h(w_{500}, b_{500}, x_1) \\ \ldots\ldots & & \ldots\ldots \\ h(w_1, b_1, x_1) & \ldots\ldots & h(w_{500}, b_{500}, x_N) \end{bmatrix} \quad (10)$$

After the initial output matrix H_0 is obtained, the initial output weight $\beta^{(0)}$ is calculated:

$$\beta^{(0)} = P_0 H_0^T T_0 \quad (11)$$

where $P_0 = (H_0^T H_0)^{-1}$, $T_0 = [t_1, t_2 \ldots t_N]^{\mathrm{T}}$, and the ELM training is finished.

pSubsequently, the trained ELM model is deployed in the local clients to identify falls from ADLs. However, its performance is limited, so we manually marked the detected false alarm (the predicted label is inconsistent with the actual label) and missed alarm events and then used the marked events to update the OS-ELM and improve the algorithm's performance. The essence of updating OS-ELM is to recalculate the output weight $\beta^{(i)}$ using a single manually labeled data, and the calculation process is shown as follows:

$$P_{k+1} = P_k - \frac{P_k h_{k+1} h_{k+1}^T P_k}{1 + h_{k+1}^T P_k h_{k+1}} \quad (12)$$

$$\beta^{(k+1)} = \beta^{(k)} + P_{k+1} h_{k+1} (t_{k+1}^T - h_{k+1}^T \beta^{(k)}) \quad (13)$$

Finally, the OS-ELM algorithm is updated by output calculation β, and the '*update flag*' is updated to 1 (the initial state is 0).

TABLE 2. The data of the Sisfall dataset.

YOUNG	AGE	FALLS	ADL	ELDERLY	AGE	FALLS	ADL
SA01	26	75	79	**SE01**	71	0	59
SA02	23	75	79	**SE02**	75	0	48
SA03	19	75	79	**SE03**	62	0	59
SA04	23	75	79	**SE04**	63	0	59
SA05	22	75	79	**SE05**	63	0	59
SA06	21	75	79	**SE06**	60	75	79
SA07	21	75	79	**SE07**	65	0	59
SA08	21	75	79	**SE08**	68	0	59
SA09	24	75	79	**SE09**	66	0	59
SA10	21	75	79	**SE10**	64	0	59
SA11	19	75	79	**SE11**	66	0	59
SA12	25	75	79	**SE12**	69	0	59
SA13	22	75	79	**SE13**	65	0	59
SA14	27	75	79	**SE14**	67	0	59
SA15	25	75	79	**SE15**	64	0	58
SA16	20	75	79				
SA17	23	75	78				
SA18	23	75	79				
SA19	30	75	79				
SA20	30	73	77				
SA21	30	75	79				
SA22	19	75	79				
SA23	24	75	79				

After a period of running on the local client, the clients with '*update flag* = 1' are selected for Federated Learning. For each round, the process mainly includes the following steps: 1) Update the model parameters of all selected clients to the cloud platform; 2) Calculate the mean value using the uploaded client parameters, and the calculation formula is as follows:

$$(w_{t+1}, \beta_{t+1}, b_{t+1}) = \frac{1}{K}\sum_{k=1}^{K}(w_t^k, \beta_t^k, b_t^k) \tag{14}$$

where $w_{t+1}, \beta_{t+1}, b_{t+1}$ represent the input weight, output weight, and deviation of the OS-ELM algorithm after federated learning. 3) Send the parameters obtained to the selected client and return to the OS-ELM algorithm update stage.

IV. EXPERIMENTS

The experimental data come from the public dataset SisFall [35], which includes 23 young people (SA01 - SA23) and 15 elderly people (SE01 - SE15), as shown in Table 2. In this dataset, the acceleration information is obtained by the sensor (ADXL345) placed on the waist of the subjects. ADLs for 19 different postures and falls for 15 different postures are collected from 23 young people and one elderly person (SE06), while some postures of ADLs are collected from the remaining 14 elderly due to health limitations. In this paper, the data in Sisfall are divided into a training dataset, a young test dataset and an elderly test dataset. The training dataset consists of 1,050 falls and 1,106 ADLs from 14 young individuals (SA01 - SA14). The young test dataset is composed of 673 falls and 708 ADLs from 9 young individuals (SA15 - SA23). Finally, the elderly dataset contains 75 elderly fall samples (SE06) and 893 elderly ADLs samples (SE01 - SE15).

A. FED-ELM EXPERIMENT

The Fed-ELM experiments of mainly include three phases: 1) the initial training of ELM; 2) the online update of the OS-ELM; 3) the Federated Learning process. In the ELM training process, the ADLs and falls of 14 young people are utilized (SA01 - SA14). The data of each person contain 75 fall samples and 79 ADL samples. First, the parameters of the ELM are randomly initialized, and then the training data are used for the preliminary training of the ELM.

In the training process, the number of hidden layer neurons of the ELM and the activation function are adjusted to improve the performance of the trained algorithm. The activation functions selected in this experiment include Softplus, Sigmoid and tanh, and the selection range of the number of neurons in the hidden layer is [50,500] (train with integral multiples of 50). Then, subdivision is performed around the number of neurons corresponding to the optimal performance of each activation function. For example, when the number of hidden layer neurons is 50, the performance is optimal. the parameter range is adjusted to [10,90] (train with integral multiples of 10) to continue training. As a result, when the activation function is tanh and the number of hidden layer neurons is 80, the training performance of ELM is the best, and the accuracy reaches 95.18%.

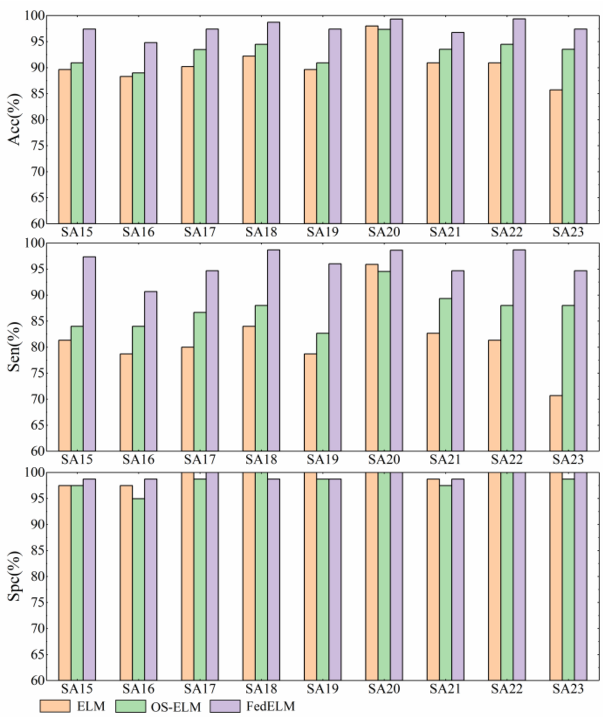


FIGURE 5. The Sen, Spc and Acc of the three algorithms (ELM, OS-ELM and Fed-ELM) on different young individuals' data.

TABLE 3. The accuracy, sensitivity and specificity of ELM, OS-ELM, and Fed-ELM on the young dataset.

Algorithm	Acc(%)	Sen(%)	Spc(%)
ELM	90.80	81.43	99.29
OS-ELM	93.41	87.22	99.01
Fed-ELM	96.96	94.50	99.29

After obtaining the trained ELM, the performance of the OS-ELM and Fed-ELM are evaluated by the fall and ADL data of 9 young people (SA15 - SA23). For Fed-ELM, each person represents a client. When a false alarm or missing alarm event is detected, the '*update flag*' of the current client is set to 1, and the parameter β of OS-ELM is updated using the manually labeled data. For the third part, the parameters are aggregated by the Federated Learning algorithm on the cloud platform, and then the updated parameters are returned to clients.

B. EXPERIMENTAL COMPARISON OF FED-ELM

To demonstrate the effectiveness of the proposed Fed-ELM algorithm, the experimental results of ELM, OS-ELM, and Fed-ELM are compared. In this paper, the performance of the three algorithms on the young test data is shown in Table 3, and the performance on different clients is also displayed in Figure 5. According to Table 3, we can see that on the whole young test dataset, the performance of ELM is the lowest, and its accuracy is only 90.80%. For OS-ELM, the accuracy on the whole test dataset is 2.6% higher than that of ELM. However, its sensitivity reaches only 87.22%, which needs to be improved. Finally, the sensitivity and specificity of the proposed Fed-ELM are improved to 94.50% and 99.29%, respectively. Fed-ELM not only performs well on the whole test dataset but also performs better on different clients. From the specificity perspective, these three algorithms perform satisfactorily. However, from the sensitivity perspective, the sensitivity of the ELM on SA23 data is only 70.67%, while the lowest sensitivity of OS-ELM and Fed-ELM is 82.67% and 90.67%, respectively. Therefore, among the three algorithms, the performance of Fed-ELM on each client is more stable. As a result, it can be concluded that Fed-ELM is better in terms of the overall performance and the stability of each client.

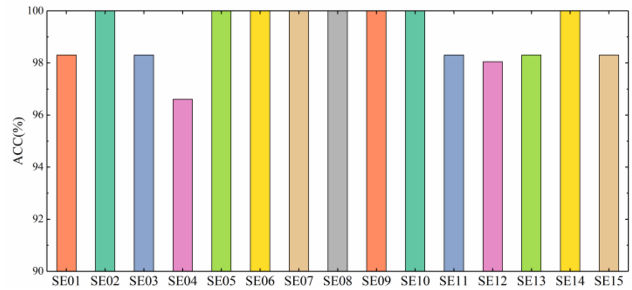


FIGURE 6. The Acc of the Fed-ELM on different data of the elderly individuals.

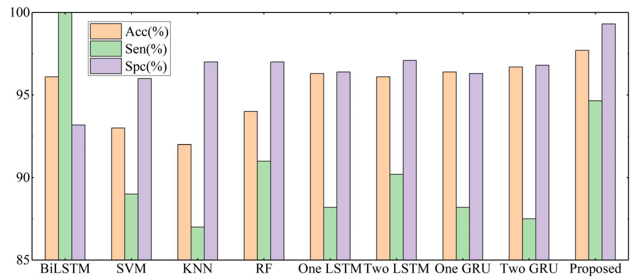


FIGURE 7. Comparison of other methods with the method proposed in this paper.

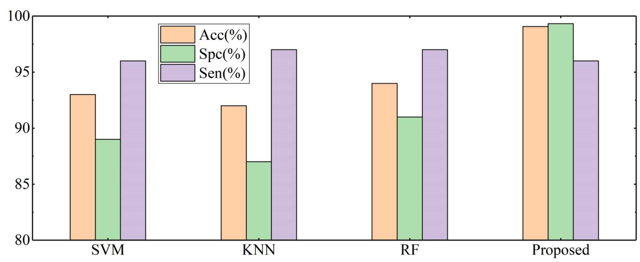


FIGURE 8. Comparison of elderly fall detection methods.

C. THE EXPERIMENT FOR THE ELDERLY AND EXPERIMENTAL COMPARISON

In this section, the elderly data in SisFall are used to test the performance of the proposed Fed-ELM algorithm (SE01 - SE15). Finally, the experimental results of Fed-ELM on each elderly individual are shown in Figure 6. Since there is no fall event in all the elderly data except SE06, only the accuracy is used to evaluate the performance on each individual. It was found that the accuracy on SE04 is the lowest, reaching 96.61%. On SE05, SE06, SE07, SE08, SE09, SE10 and SE14, all the samples are classified correctly. It can be concluded that the performance of the proposed Fed-ELM on different elderly users is satisfactory. In addition to analyzing the accuracy of different users, the data of SE06 are used to test the sensitivity and specificity obtained by the Fed-ELM. Ultimately, the sensitivity and specificity achieved on SE06 were 100% and 96%, respectively. As a result, the proposed Fed-ELM can effectively distinguish falls from ADLs among elderly individuals.

D. DISCUSSION

To prove the effectiveness of the proposed Fed-ELM, the proposed method is compared with the existing mainstream fall detection methods. In [36], [37], and [38], BiLSTM, SVM, KNN, RF and four recurrent neural network methods were used to identify falls and ADLs, and all the datasets used in these experiments were SisFall datasets. In Figure 6, the performance of the proposed Fed-ELM and the eight fall detection algorithms in [36], [37] and [38] are compared. As shown in Figure 7, among the 9 methods, the accuracy of the two GUR layers reaches 96.7%. Nevertheless, its sensitivity is lower than 90%, meaning that it is prone to leakage alarm.. In terms of sensitivity, the sensitivity of BiLSTM reached 100%, but the specificity was only 93.18%. In terms of specificity, the method proposed in this paper has the best performance, with a specificity of 97.70%. It can be concluded that the performance of the proposed Fed-ELM is satisfactory, and can better balance the sensitivity and specificity.

In addition, the experiments in [37] classify the elderly data in SisFall with three methods: SVM, KNN, and RF. As Figure 8 shows, in distinguishing falls from ADLs of elderly individuals, the sensitivity, specificity and accuracy of the Fed-ELM proposed in this paper are all better than those of the three methods proposed in [37]. In [37], the highest accuracy, specificity and sensitivity in the elderly were 94%, 97% and 91%, respectively. On the same test dataset, the accuracy, sensitivity and specificity of the proposed Fed-ELM achieve 98.76%, 96% and 99.07%, respectively. In conclusion, Fed-ELM performs excellently in distinguishing elderly falls from ADLs.

V. CONCLUSION

In this paper, a fall detection method based on Federated Learning and Extreme Learning Machine is proposed. Due to inconsistencies in age, height and weight, the gaits of different individuals are different. Furthermore, the accuracy of the trained fall detection algorithm in distinguishing actual user falls from daily activities decrease. In this case, using the users' data to update the trained fall detection algorithm can improve the detection accuracy. In addition, the data of a single user are limited, and the Federated Learning algorithm can obtain the useful information of all users without involving private data. The experimental results show that, directly compared with the application of the trained algorithm, the accuracy of the fall detection algorithm based on the proposed Fed-ELM for the young individuals' data is increased by 6.16% compared with directly using the trained ELM. In addition, the performance of Fed-ELM on the elderly individuals' data is also excellent, with the accuracy rate exceeding 96% on different users.

However, the data used in this paper were collected in the experimental environment and could not include all the real world motion gestures. In addition, the algorithm proposed in this paper requires users to manually mark the wrong data. This approach is prone to mislabeling, which can lead to a reduction in the quality of Fed-ELM. In the future, we plan to conduct experiments using falls and daily activity data in real life situations to further improve the performance of the algorithm in real applications. In addition, the quality of Federated Learning would be improved by increasing the defense of Federated Learning to avoid performance losses caused by incorrect data labeling.

ACKNOWLEDGMENT

(Zhigang Yu and Jiahui Liu contributed equally to this work.)

REFERENCES

[1] C. Todd and D. Skelton, *What Are the Main Risk Factors for Falls Amongst Older People and What Are the Most Effective Interventions to Prevent These Falls?* Geneva, Switzerland: World Health Organization, 2004.

[2] M. E. Tinetti, "Predictors and prognosis of inability to get up after falls among elderly persons," *J. Amer. Med. Assoc.*, vol. 269, no. 1, pp. 65–70, Jan. 1993.

[3] E. E. Geertsema, G. H. Visser, M. A. Viergever, and S. N. Kalitzin, "Automated remote fall detection using impact features from video and audio," *J. Biomechanics*, vol. 88, pp. 25–32, May 2019.

[4] M. G. Amin, Y. D. Zhang, F. Ahmad, and K. C. D. Ho, "Radar signal processing for elderly fall detection: The future for in-home monitoring," *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 71–80, Mar. 2016.

[5] N. Faulkner, B. Parr, F. Alam, M. Legg, and S. Demidenko, "CapLoc: Capacitive sensing floor for device-free localization and fall detection," *IEEE Access*, vol. 8, pp. 187364–787363, 2022.

[6] M. Turan and B. Barshan, "Classification of fall directions via wearable motion sensors," *Digit. Signal Process.*, vol. 125, Jun. 2022, Art. no. 103129.

[7] B.-S. Lin, T. Yu, C.-W. Peng, C.-H. Lin, H.-K. Hsu, I.-J. Lee, and Z. Zhang, "Fall detection system with artificial intelligence-based edge computing," *IEEE Access*, vol. 10, pp. 4328–4339, 2022.

[8] D. Mrozek, A. Koczur, and B. Maysiak-Mrozek, "Fall detection in older adults with mobile IoT devices and machine learning in the cloud and on the edge," *Inf. Sci.*, vol. 537, pp. 132–147, Oct. 2020.

[9] V. Kumar, N. Badal, and R. Mishra, "Elderly fall due to drowsiness: Detection and prevention using machine learning and IoT," *Modern Phys. Lett. B*, vol. 35, no. 7, Mar. 2021, Art. no. 2150120.

[10] A. De, A. Saha, P. Kumar, and G. Pal, "Fall detection approach based on combined two-channel body activity classification for innovative indoor environment," *J. Ambient Intell. Humanized Comput.*, pp. 1–12, Jan. 2022.

[11] F. Shu and J. Shu, "An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost Android box," *Sci. Rep.*, vol. 11, no. 1, p. 2471, Jan. 2021.

[12] G. Feng, J. Mai, Z. Ban, X. Guo, and G. Wang, "Floor pressure imaging for fall detection with fiber-optic sensors," *IEEE Pervasive Comput.*, vol. 15, no. 2, pp. 40–47, Jun. 2016.

[13] A. K. Bourke and G. M. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," *Med. Eng. Phys.*, vol. 30, no. 1, pp. 283–292, Jul. 2019.

[14] Y. Wu, Y. Su, R. Feng, N. Yu, and X. Zang, "Wearable-sensor-based pre-impact fall detection system with a hierarchical classifier," *Measurement*, vol. 140, pp. 84–90, Jan. 2008.

[15] C. Wang, M. R. Narayanan, S. R. Lord, S. J. Redmond, and N. H. Lovell, "A low-power fall detection algorithm based on triaxial acceleration and barometric pressure," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2014, pp. 570–573.

[16] C. Wang, W. Lu, M. Narayanan, D. Chang, S. Lord, S. Redmond, and N. Lovell, "Low-power fall detector using triaxial accelerometry and barometric pressure sensing," *IEEE Trans. Ind. Informat.*, vol. 12, no. 6, pp. 2302–2311, Dec. 2016.

[17] W. Lu, C. Wang, M. C. Stevens, S. J. Redmond, and N. H. Lovell, "Low-power operation of a barometric pressure sensor for use in an automatic fall detector," in *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2016, pp. 2010–2013.
[18] M. C. Stevens, W. Lu, C. Wang, S. J. Redmond, and N. H. Lovell, "A Kalman filter to estimate altitude change during a fall," in *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2016, pp. 5889–5921.
[19] C. Wang, S. Redmond, W. Lu, M. Stevens, S. Lord, and N. Lovell, "Selecting power-efficient signal features for a low-power fall detector," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 11, pp. 2729–2736, Nov. 2017.
[20] W. Lu, M. C. Stevens, C. Wang, S. J. Redmond, and N. H. Lovell, "Smart triggering of the barometer in a fall detector using a semi-permeable membrane," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 1, pp. 146–157, Jan. 2020.
[21] N. Zurbuchen, A. Wilde, and P. Bruegger, "A machine learning multi-class approach for fall detection systems based on wearable sensors with a study on sampling rates selection," *Sensors*, vol. 21, no. 3, p. 938, Jan. 2021.
[22] J. A. Santoyo-Ramón, E. Casilari, and J. M. Cano-García, "A study of the influence of the sensor sampling frequency on the performance of wearable fall detectors," *Measurement*, vol. 193, Apr. 2022, Art. no. 110945.
[23] Y. H. Nho, S. Ryu, and D. S. Kwon, "UI-GAN: Generative adversarial network-based anomaly detection using user initial information for wearable devices," *IEEE Sensors J.*, vol. 21, no. 8, pp. 9949–9958, Apr. 2021.
[24] M. J. Al Nahian, T. Ghosh, M. H. Al Banna, M. A. Aseeri, M. N. Uddin, and M. R. Ahmed, "Towards an accelerometer-based elderly fall detection system using cross-disciplinary time series features," *IEEE Access*, vol. 9, pp. 39413–39431, 2021.
[25] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.
[26] J. Xu, B. S. Glicksberg, C. Su, P. Walker, and J. Bian, "Federated learning for healthcare informatics," *J. Healthc Inform. Res.*, vol. 5, pp. 1–19, Dec. 2021.
[27] N. Rieke, J. Hancox, W. Li, F. Milletarì, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, S. Ourselin, M. Sheller, R. M. Summers, A. Trask, D. Xu, M. Baust, and M. J. Cardoso, "The future of digital health with federated learning," *NPJ Digit. Med.*, vol. 3, no. 1, p. 119, Sep. 2020.
[28] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.
[29] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1342–1397, 2nd Quart., 2021.
[30] McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Art. Intel. Statist.*, 2017, pp. 1273–1282.
[31] D. Mrozek, A. Koczur, and B. Maysiak-Mrozek, "Fall detection in older adults with mobile IoT devices and machine learning in the cloud and on the edge," *Inf. Sci.*, vol. 537, pp. 132–147, Oct. 2020.
[32] K. Liu, C.-Y. Hsieh, S. J.-P. Hsu, and C.-T. Chan, "Impact of sampling rate on wearable-based fall detection systems based on machine learning models," *IEEE Sensors J.*, vol. 18, no. 23, pp. 9882–9890, Dec. 1, 2018.
[33] T. Xu, J. Liu, and M. Geng, "Fall detection method based on TBM and SVM," in *Proc. 41st Chin. Control Conf. (CCC)*, Jul. 2022, pp. 2984–2989.
[34] T. Xu and J. Liu, "A low-power fall detection method based on optimized TBM and RNN," *Digit. Signal Process.*, vol. 126, Jun. 2022, Art. no. 103525.
[35] A. Sucerquia, J. D. Lòpez, and J. F. Vargas-Bonilla, "Sisfall: A fall and movement dataset," *Sensors J.*, vol. 17, no. 1, p. 198, 2017.
[36] M. Waheed, H. Afzal, and K. Mehmood, "NT-FDS—A noise tolerant fall detection system using deep learning on wearable devices," *Sensors*, vol. 21, no. 6, p. 2006, Mar. 2021.
[37] J. Alizadeh, M. Bogdan, J. Classen, and C. Fricke, "Support vector machine classifiers show high generalizability in automatic fall detection in older adults," *Sensors*, vol. 21, no. 21, p. 7166, Oct. 2021.
[38] E. Torti, A. Fontanella, M. Musci, N. Blago, D. Pau, F. Leporati, and M. Piastra, "Embedded real-time fall detection with deep learning on wearable devices," in *Proc. 21st Euromicro Conf. Digit. Syst. Design (DSD)*, Aug. 2018, pp. 405–412.

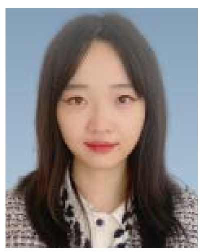
ZHIGANG YU received the joint M.S. degree in computer technology from the North University of China and University of Chinese Academy of Sciences, in 2018. He is currently a Research Staff with the Institute of Big Data and Artificial Intelligence, China Telecom Corporation Ltd., Beijing Research Institute, Beijing, China. His current research interests include computer vision, AI security, and federated learning.

JIAHUI LIU was born in Liaoning, China, in 1997. She received the B.Sc. and M.Sc. degrees from Shenyang Aerospace University, China, in 2019 and 2022, respectively. Her current research interest includes fall detection.

MINGCHUAN YANG received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2002. He is currently the Director of the Institute of Big Data and Artificial Intelligence, China Telecom Research Institute. His research interests include big data and artificial intelligence.

YANMIN CHENG received the M.S. degree from Beijing Jiaotong University, Beijing, China, in 2018. He is currently working as a Research and Development Project Manager at the China Telecom Research Institute. His research interests include federated learning, AI security, and NLP techniques.

JIE HU received the M.S. degree in software engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2017. Her research interests include information retrieval, multi-hop reasoning, and natural language processing.



XINCHI LI received the M.S. degree in computer application from the Chinese Academy of Sciences. She is mainly engaged in technical research and product development in the fields of big data and blockchain. She has published more than ten papers and written two monographs.