A Coarse-to-Fine Feature Selection Method for Accurate Detection of Cerebral Small Vessel Disease

Yiqiang Chen [a,b],Meiyu Huang [a,b], Chunyu Hu [a,b] ,Yicheng Zhu [c], Fei Han [c], Chunyan Miao[d]
[a] Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China 100190
[b] University of Chinese Academy of Sciences, Beijing, China 100190
[c] Peking Union Medical College Hospital, Beijing, China 100730
[d] Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore

***Abstract*—Cerebral small vessel disease (SVD) is common in the elderly and is associated with loss of functional independence, institutionalization, and death. In this paper, we propose a coarse-to-fine feature selection method for accurate SVD detection and timely implementation of interventions. The proposed method first uses an Iterative Random Forest based Feature Selection (IRFFS) method to obtain the most representative features from a feature set that includes gait, balance, and agility performance features extracted from 17 predefined clinical actions. The method then uses the Feature Incremental Extreme Learning Machine (FIELM) model to further verify the discriminant ability of each kind of selected features. Our results demonstrate that the proposed method can effectively select the most significant features for SVD detection, which include gait and agility performance features. Our method achieves up to 91.44% classification accuracy, outperforming other state-of-the-art feature selection methods. Our findings also verify clinical observations indicating that the fine motor pattern features of upper and lower limbs are helpful for high-accuracy SVD detection.**

Keywords—coarse-to-fine feature selection; detection of cerebral small vessel disease; iterative random forest-based feature selection; feature incremental extreme learning machine

I. INTRODUCTION

Cerebral small vessel disease (SVD), which includes white matter lesions (WML) and lacunar infarcts, is frequently observed on Magnetic Resonance Imaging (MRI) scans of the elderly. It is widely accepted that SVD is a main cause of motor function decline and is often associated with loss of functional independence, hospitalization, and death, representing a major challenge for public health. Accurate early detection of SVD allows for timely implementation of interventions, thus mitigating the negative impacts on the patient. Previous studies have shown the importance of measuring an elderly person's gait and balance for detection of SVD [1] or white matter changes [2]. However, at an early stage, gait disturbances may be solely manifested as normal to mild slowing and subjective postural instability [3], which is not easily distinguishable from age-related decline even for trained specialists. In addition to gait and balance dysfunction, fine movement disorders of the upper and lower limbs have also been observed in clinical practice, but up to now evidence from clinical research is still lacking.

In this paper, we describe our studies investigating the use of motor pattern features for accurate SVD detection. These features include gait, balance, and agility performance features extracted from 17 predefined clinical actions. In order to obtain the most significant motor pattern features for distinguishing SVD patients from the healthy elderly, we propose a coarse-to-fine feature selection method. This method first employs an Iterative Random Forest based Feature Selection (IRFFS) method to determine the most representative features from the original feature set. The method then uses the Feature Incremental Extreme Learning Machine (FIELM) model to further verify the discriminant ability of each kind of selected features. Experimental results demonstrate that the IRFFS method is effective in feature selection and performs better than other state-of-the-art feature selection methods provided in WEKA [4]. Besides, the classification results from FIELM verify clinical observations indicating that fusing gait and agility related motor pattern features is helpful for building a more accurate SVD detection model.

II. RELATED WORK

Public health issues such as the global population aging phenomenon have attracted considerable attention recently. Recent advances in computer science have made possible various research works looking into the combination of digital medicine and machine learning techniques. For example, M. Termenon et al. [5] built a brain MRI morphological patterns extraction tool based on the Extreme Learning Machine (ELM) method [6] and majority vote classification. Chen et al. proposed an accurate b-COELM [7] based approach for automating stroke detection through a computer-based, body sensing game-based Trail Making Test [8]. A common challenge in these disease detection problems is how to select the most discriminant features for high-accuracy detection performance. Existing feature selection methods [4] are mainly focused on big data, and are thus not suitable for those disease detection problems with only a small number of available samples. Addressing this issue, we introduce the IRFFS method, which aims to select the most significant features from small-sampling data of high-dimensionality motor pattern features. Furthermore, in order to determine the type of motor pattern features most related to SVD, we employ the

978-1-5090-0620-5/16/$31.00

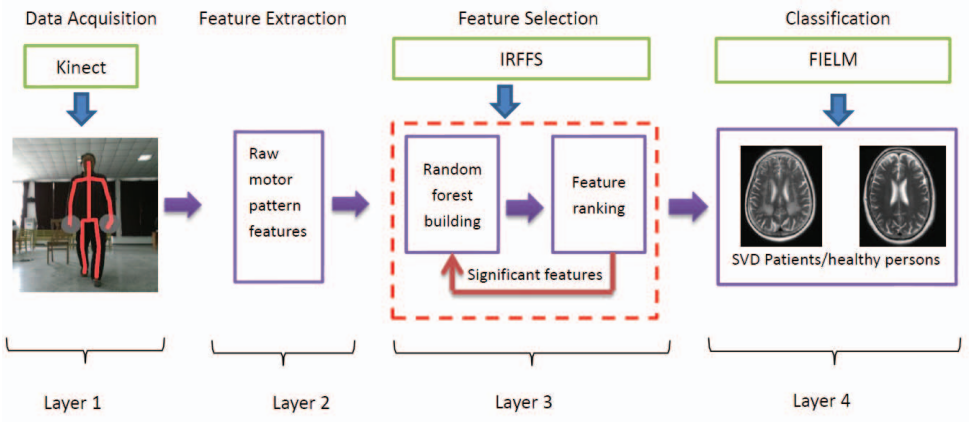

Fig. 1. The proposed SVD detection process.

FIELM model, which can transfer an original ELM classification model to a new one with incremental features in a unified framework. Developed by Huang et al, ELM belongs to the artificial neural network (ANN) family, and more specifically to the Single Layer Feedforward Network (SLFN) subfamily [6]. In ELM, the learning step is made without iterative tuning and is very suitable for incremental learning [9-10]. With the help of FIELM, doctors are able to distinguish SVD patients from normal control subjects with gradually increased features of different types and can identify the most important ones for accurate SVD detection.

III. MOTOR PATTERN FEATURE SELECTION FOR SVD DETECTION

As shown in Fig. 1, the proposed SVD detection process consists of four layers: 1) data acquisition, in which raw motion data is collected by the depth sensor and the RGB camera embedded in the Kinect; 2) feature extraction, which extracts all potential motor pattern features related to SVD using the raw motion data from layer 1; 3) feature selection, where we use the IRFFS method to obtain the most representative features; and 4) classification, which uses FIELM to validate the discriminant ability of the selected features.

A. Data Acquisition

The first layer uses Kinect to collect raw depth sensor data and RGB camera data of 17 predefined clinical actions for each subject. Specifically, the 17 actions are natural walking, tandem walking (heels to toes), 3-meter walk test and turning 180°, side by side stand, semi-tandem stand, tandem stand, nudge (light push on sternum, subject with feet close together), pull test (response to sudden, strong posterior displacement produced by pull on shoulders), tapping heel on the ground in rapid succession picking up entire leg, natural chair stand, chair stand with arms folded across the chest, repeated chair stand with arms folded across the chest, finger chase, finger nose test, rapid alternating movements of the hand (10 cycles of repetitive alternation of pronation and supination movements of the hand on his/her thigh as fast and precise as possible), rapid alternating movements of hands (pronation-supination movements of both hands simultaneously), and finger tapping. These actions are summarized and simplified according to several existing and widely accepted scales, such as the Scale for Assessment and Rating of Ataxia (SARA), the Short Physical Performance Battery (SPPB), the Tinetti Mobility Test, and the Unified Parkinson Disease Rating Scale (UPDRS). The RGB camera data is used for action segmentation and labeling, and the depth sensor data is used for obtaining the coordination of skeleton joints of each action using the pose tracking method proposed by Shotton et al. [11].

B. Feature Extraction

Layer 2 extracts all potential SVD related motor pattern features from the coordination data collected in layer 1. These features include the following: 1) gait related features, such as stride length, step width, step height, walking velocity, and sagittal and coronal-angular excursions of the shoulder, elbow, hip, knee and trunk, which are calculated as the method in Zijlmans et al. [12]; 2) balance related features, such as angular excursions of the trunk and stand velocity; and 3) agility related features, including smoothness of moving trajectory and variability of moving velocity. In total, 739 features are extracted.

C. Feature Selection

To determine the most representative features for SVD detection, we use the IRFFS method in layer 3. This method automatically measures the importance of the 739 features, retaining the most representative features and discarding unnecessary ones.

1) Random forests

Random forests were first introduced in a paper by Leo Breiman [13]. This paper describes a method for building a forest of uncorrelated decision trees [14] using a Classification And Regression Tree (CART) [15] procedure, combined with randomized node optimization and bagging.

a) Decision tree learning

Decision tree learning refers to the construction of a decision tree from class-labelled training tuples [14]. A decision tree is a flowchart-like structure where each internal node denotes a test on a feature that best splits the training set using Gini impurity or information gain [16], each branch represents the outcome of a test, and each leaf node holds a class label. Decision trees are a popular method for various machine learning tasks because it is invariant under scaling and various other transformations of feature values. However, trees that are grown very deep tend to learn highly irregular patterns: they overfit their training sets because they have low bias, but very high variance. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance of the final model.

b) Tree bagging

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set $X = x_1, \cdots, x_n$ with responses $Y = y_1, \cdots, y_n$, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples.

Specifically, for $b = 1, \cdots, B$:

1. Sample n training examples from X, Y with replacement, and call these examples X_b, Y_b.

2. Train a decision or regression tree f_b on X_b, Y_b.

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

or by taking the majority vote in the case of classification trees. This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic); bootstrap sampling is a way of de-correlating the trees by showing them different training sets. The number of trees, B, is a free parameter which can be optimized using cross-validation. The training and test errors tend to level off after some number of trees have been fit.

c) From bagging to random forests

The above procedure describes the original bagging algorithm for trees. Random forests differ in only one way from this general scheme: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features. This process is sometimes called "feature bagging". The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the response variable (target output), these features will be selected in many of the B trees, causing them to become correlated. An analysis of how bagging and random subspace projection contribute to accuracy gains under different conditions is given by Ho [17]. Typically, for a classification problem with n features, $\left\lfloor \sqrt{n} \right\rfloor$ features are used in each split. For regression problems the inventors recommend $\left\lfloor n/3 \right\rfloor$ with a minimum node size of 5 as the default.

2) Iterative Random Forest-based Feature Selection

In [15], Leo Breiman proposes to use random forests for feature selection by ranking the importance of features through permutation. More specifically, to measure the importance of the j^{th} feature after training, the values of the j^{th} feature are permuted among the training data and the out-of-bag error is again computed on this perturbed data set. The importance score for the j^{th} feature is computed by averaging the difference in out-of-bag error before and after the permutation over all trees. The score is normalized by the standard deviation of these differences. Features which produce large values for this score are ranked as more important than features which produce small values. This method of determining variable importance works well for big data; however, it is not suitable for small-sampling data because for such data, for each selected feature, even a little permutation would cause a large difference in out-of-bag error. In order to avoid this problem, we introduce the IRFFS method to select the most significant features from small-sampling data with high-dimensional features. This method measures the importance of each feature according to its occurrence frequencies in the trained random forests, motivated by the fact that features with very strong discriminant ability for the response variable (target output) will be selected in many trees of the trained random forests.

Specifically, IRFFS is an iterative loop process. Initially, we use all the 739 features to build several random forests using the learning method proposed by Breiman et al. [15] based on repeatedly sampled training datasets. Then, at each loop, we rank the features according to their occurrence frequencies in the built random forests and select the most common and frequent features to rebuild new random forests. We repeat the process until no further classification performance improvements can be obtained. Let us denote the initial training set as $T = (\mathbf{x}_i, \mathbf{t}_i) \in R^n \times R^m, i = 1, 2, \cdots, N$, where $\mathbf{x}_i = \left[\mathrm{x}_{i1}, \mathrm{x}_{i2}, \cdots, \mathrm{x}_{in}\right]^{\mathrm{T}}$ is an input vector with n features and $\mathbf{t}_i = \left[\mathrm{t}_{i1}, \mathrm{t}_{i2}, \cdots, \mathrm{t}_{im}\right]^{\mathrm{T}}$ is the corresponding target vector. The main procedure of IRFFS can be described as follows:

Step 1. Initialization. Set iterative index $t = 0$, initial training set $T(0) = T$, initial average classification accuracy $\overline{acc} = 0$.

Step 2. Random sampling. Randomly sample N_r samples from the training set $T(t)$ for S times, obtaining S sub-training sets $T^s(t)=\{\mathbf{x}_i,\mathbf{t}_i\}\in R^{n(t)}\times R^m, i=1,2,\cdots,N_r$ and S sub-testing sets $V^s(t)=\{\mathbf{x}_i,\mathbf{t}_i\}\in R^{n(t)}\times R^m, i=N_r+1,N_r+2,\cdots,N$, $s=1,2,\cdots,S$.

Step 3. Random forest building. Build S random forests based on S sub-training sets $T^s(t)$ using the learning method proposed in [13]. Here, the number of trees in each random forest is set to B, and $\left\lfloor\sqrt{n(t)}\right\rfloor$ features are used in each split.

Step 4. Feature validation. Compute the average classification accuracy $\overline{acc}(t)$ of the S built random forests using the corresponding sub-testing sets $V^s(t)$. If $\overline{acc}(t)$ is higher than $\overline{acc}$, set $\overline{acc}$ to $\overline{acc}(t)$ and go to Step 5, else drop the iterative process and select the current $n(t)$ features as the most significant features.

Step 5. Feature sorting. In order to measure the importance of each feature f_j, $j=1,2,\cdots,n(t)$, we introduce two variables: 1) the number of built random forests where f_j occurs, denoted as N_j, and 2) the frequency at which f_j appears in the built random forests, denoted as F_j. N_j can be counted directly, and F_j is computed as follows:

$$F_j=\sum_{s=1}^{S}\sum_{b=1}^{B}\frac{N_{bj}^s}{N_b^s},$$

where N_{bj}^s and N_b^s denote the number of internal nodes testing on feature f_j and the number of total internal nodes in the b^{th} tree of the s^{th} built random forest, respectively. Since a more discriminant feature would be selected in more trees of more random forests built using different training sets, a feature with large values of N_j and F_j is of greater importance. We then sort the $n(t)$ features according to N_j and F_j in descending order.

Step 6. Feature selection. Select the top r ranked features as the most discriminant features. Since a large r would cause redundant looping, while a small r may discard significant features, we set r as half of the current number of features, i.e. $r=\left\lfloor n(t)/2\right\rfloor$. If r is larger than the minimum number of features $n_{\min}$, assign $n(t+1)$ to r and go to Step 7, else drop the iterative process and select the current $n(t)$ features as the most significant features.

Step 7. Loop. Set $t=t+1$, and go to Step 2.

D. Classification

To further validate the discriminant ability of each kind of selected features in Layer 3, we introduce the FIELM method in Layer 4. FIELM is a learning method built on ELM which can transfer an original ELM model to a new one with incremental features in a unified framework. In this transformation process, FIELM can add new features to the training model without changing the existing structure. With the help of FIELM, doctors are able to distinguish SVD patients from normal control subjects with gradually increased features of different types and can identify the most important ones for accurate SVD detection.

1) ELM

Developed by Huang, et al, ELM belongs to the ANN family, and more specifically to the SLFN subfamily [6]. In ELM, learning is made without iterative tuning and is very efficient and effective when the training set is small [18]. According to ELM learning theory, if SLFNs $\mathrm{f}(\mathbf{x})=\mathrm{h}(\mathbf{x})\beta$, with tunable piecewise continuous hidden-layer feature mapping $\mathrm{h}(\mathbf{x})$, can approximate any target continuous functions, tuning is not required in the hidden layer [19]. All the hidden-node parameters, which are supposed to be tuned by conventional learning algorithms, can be randomly generated according to any continuous sampling distribution [20]. Compared with other traditional learning methods, ELM has not only better performance in classification precision and regression fitting degree, but is also more time-efficient in offline learning and online prediction [9-10].

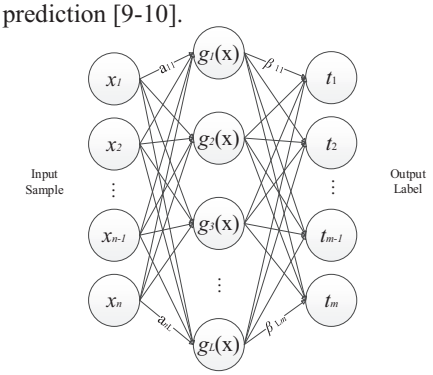

Fig. 2. SLFN with *L* hidden nodes

Given N arbitrary distinct samples $(\mathbf{x}_i,\mathbf{t}_i)\in R^n\times R^m, i=1,2,\cdots,N$, where $\mathbf{x}_i=\left[\mathrm{x}_{i1},\mathrm{x}_{i2},\cdots,\mathrm{x}_{in}\right]^{\mathrm{T}}$ is an input vector with n features and $\mathbf{t}_i=\left[\mathrm{t}_{i1},\mathrm{t}_{i2},\cdots,\mathrm{t}_{im}\right]^{\mathrm{T}}$ is the corresponding target vector, the network with L hidden nodes is shown in Fig. 2.

The output function of this network can be represented as follows:

$$f_L(\mathbf{x}_i)=\sum_{l=1}^{L}\boldsymbol{\beta}_l G(\mathbf{a}_l,b_l,\mathbf{x}_i),i=1,\cdots,N, \tag{1}$$

where $\mathbf{a}_l$ and b_l are the learning parameters of hidden nodes, and $\boldsymbol{\beta}_l$ is the weight connecting the l^{th} hidden node to the output node. $G(\mathbf{a}_l,b_l,\mathbf{x})$ is the output of the l^{th} hidden node with respect to the input $\mathbf{x}$. For an additive hidden node with activation function $g(\mathrm{x}):\mathrm{R}\to\mathrm{R}$ (e.g. sigmoid and threshold), $G(\mathbf{a}_l,b_l,\mathbf{x})$ is given by

$$G(\mathbf{a}_l,b_l,\mathbf{x})=g(\mathbf{a}_l\cdot\mathbf{x}+b_l),b_l\in R. \tag{2}$$

If a SLFN with L hidden nodes can approximate these N samples with zero error, it then implies that there exist $\boldsymbol{\beta}_l,\mathbf{a}_l$ and b_l such that

$$f_L(\mathbf{x}_i)=\sum_{l=1}^{L}\beta_l G(\mathbf{a}_l,b_l,\mathbf{x}_i)=\mathbf{t}_j,i=1,\cdots,N. \tag{3}$$

Equation (3) can be summarized as

$$\mathbf{H}\boldsymbol{\beta}=\mathbf{T} \tag{4}$$

where

$$\mathbf{H}(\mathbf{a}_1,\cdots,\mathbf{a}_L,\mathrm{b}_1,\cdots,\mathrm{b}_L,\mathbf{x}_1,\cdots,\mathbf{x}_N)=\begin{bmatrix}G(\mathbf{a}_1,b_1,\mathbf{x}_1)&\cdots&G(\mathbf{a}_L,b_L,\mathbf{x}_1)\\ \vdots&\ddots&\vdots\\ G(\mathbf{a}_1,b_1,\mathbf{x}_N)&\cdots&G(\mathbf{a}_L,b_L,\mathbf{x}_N)\end{bmatrix} \tag{5}$$

$$\boldsymbol{\beta}=\begin{bmatrix}\boldsymbol{\beta}_1^{\mathrm{T}}\\ \vdots\\ \boldsymbol{\beta}_L^{\mathrm{T}}\end{bmatrix}_{L\times m}\quad\text{and}\quad\mathbf{T}=\begin{bmatrix}\mathbf{t}_1^{\mathrm{T}}\\ \vdots\\ \mathbf{t}_N^{\mathrm{T}}\end{bmatrix}_{N\times m}. \tag{6}$$

According to [21], the hidden node parameters $\mathbf{a}_l$ and b_l (input weights and biases or centers and impact factors) of SLFNs do not need to be tuned during training and may simply be assigned random values. The smallest norm least-squares solution of the above linear system is:

$$\hat{\boldsymbol{\beta}}=\mathbf{H}^{\dagger}\mathbf{T}, \tag{7}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of matrix $\mathbf{H}$ [22]. Different methods can be used to calculate the Moore–Penrose generalized inverse of a matrix: the orthogonal projection method, the orthogonalization method, the iterative method, and singular value decomposition [23]. The orthogonal projection method [23] can be used in two cases: when $\mathbf{H}^T\mathbf{H}$ is nonsingular and $\mathbf{H}^{\dagger}=(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$, or when $\mathbf{H}\mathbf{H}^T$ is nonsingular and $\mathbf{H}^{\dagger}=\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}$.

2) Feature Incremental Extreme Learning Machine

In order to evaluate the discriminant ability of each kind of selected features in an incremental fashion, we propose the FIELM method, which can update the model using the training data with new increased features.

Given N arbitrary distinct samples $(\mathbf{x}_i,\mathbf{t}_i)\in R^n\times R^m,i=1,2,\cdots,N$, where $\mathbf{x}_i$ is a $n\times1$ input vector $\mathbf{x}_i=[\mathrm{x}_{i1},\mathrm{x}_{i2},\cdots,\mathrm{x}_{in}]^{\mathrm{T}}$ and $\mathbf{t}_i$ is a $m\times1$ target vector $\mathbf{t}_i=[\mathrm{t}_{i1},\mathrm{t}_{i2},\cdots,\mathrm{t}_{im}]^{\mathrm{T}}$, according to Eq. (7), when given an incremental $n1$ features, i.e. when changing the feature dimension from n to $n'=n+n1$, the learned ELM model can be updated as follows:

$$\boldsymbol{\beta}_1=\mathbf{H}_1^{\dagger}\mathbf{T} \tag{8}$$

$$\mathbf{H}_1=\begin{bmatrix}G(\mathbf{a}_1',b_1,\mathbf{x}'_1)&\cdots&G(\mathbf{a}_L',b_L,\mathbf{x}'_1)\\ \vdots&\ddots&\vdots\\ G(\mathbf{a}_1',b_1,\mathbf{x}'_N)&\cdots&G(\mathbf{a}_L,b_L,\mathbf{x}'_N)\end{bmatrix} \tag{9}$$

where $\mathbf{x}_i'=[\mathrm{x}_{i1}',\mathrm{x}_{i2}',\cdots,\mathrm{x}_{in'}']^{\mathrm{T}}$, $\mathbf{a}_l'$ is the weight vector connecting the input layer to the l^{th} hidden node and b_l is the bias of the l^{th} hidden node.

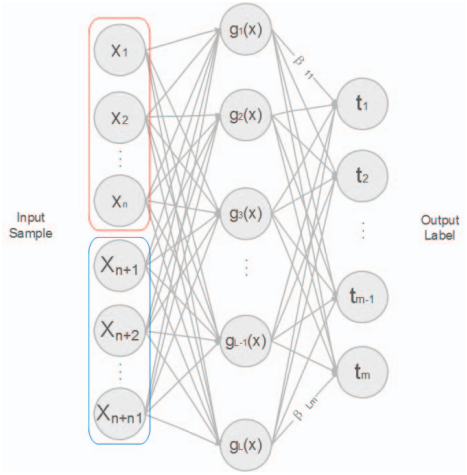

Fig. 3. FIELM Network

As shown in Fig. 3, when the feature dimension is changed, the bone structure of the network does not change. Therefore, the number of hidden nodes L, the activation function $g(\mathrm{x}):\mathrm{R}\to\mathrm{R}$, and the bias b_l should not be

changed. However, since the feature dimension is different from previously, we have to adjust $\mathbf{a}_l$ to $\mathbf{a}_l^{'}$ to accommodate the new feature dimension. To do so, we propose an input-weight transformation matrix $\mathbf{P}$, and an input-weight supplement vector $\mathbf{Q}_l$ to generate $\mathbf{a}_l^{'}$ by equation (10):

$$\left\{\mathbf{a}_l^{'} = \mathbf{a}_l \cdot \mathbf{P} + \mathbf{Q}_l\right\}_{l=1}^{L} \tag{10}$$

where

$$\mathbf{P} = \begin{bmatrix} P_{11} & \cdots & P_{1n'} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{nn'} \end{bmatrix}_{n \times n'} \tag{11}$$

$$\mathbf{Q}_l = \begin{bmatrix} Q_1 & \cdots & Q_{n'} \end{bmatrix}_{1 \times n'} \tag{12}$$

Matrix $\mathbf{P}$ has the following rules:

- Each line has only one '1', and the rest are all '0';
- Each column has one '1' at most, and the rest are all '0';
- If $P_{ij} = 1$, it means that after the feature dimension change, the i^{th} dimension of the original feature vector has become the j^{th} dimension of the new feature vector.

$\mathbf{Q}_l$ is used to account for the fact that when the feature dimension increases, we need to add the corresponding input weight for the newly added features. $\mathbf{Q}_l$ has the following rules:

- If the i^{th} item of $\mathbf{a}_i^{'}$ is an original feature, the i^{th} item of $\mathbf{Q}_l$ is '0'.
- If the i^{th} item of $\mathbf{a}_i^{'}$ is a new feature, the i^{th} item of $\mathbf{Q}_l$ should be generated randomly according to the distribution of $\mathbf{a}_i$.

FIELM can be summarized in the following steps:

Step 1. Determine the model parameters using the original dataset of N samples, such as the number of hidden nodes L and the activation function g(x).

Step 2. Randomly assign the values of weight vectors $\mathbf{a}_l$ and bias scalars $b_l, l = 1, 2, \cdots, L$.

Step 3. Calculate the original hidden layer output matrix $\mathbf{H}$.

Step 4. Calculate the initial model parameter $\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T}$.

Step 5. When adding $n1$ new features, generate the input-weight transformation matrix $\mathbf{P}$, and the input-weight supplement vector $\mathbf{Q}_l$, $l = 1, 2, \cdots, L$ according to rules mentioned above.

Step 6. Calculate the new weight vector $\mathbf{a}_l^{'} = \mathbf{a}_l \cdot \mathbf{P} + \mathbf{Q}_l$, $l = 1, 2, \cdots, L$.

Step 7. Use the new weight vector $\mathbf{a}_i^{'}$ to calculate the model parameter $\mathbf{H}_1$ using equation (9).

Step 8. Calculate $\boldsymbol{\beta}_1$ using equation (8).

Step 9. Keep performing Steps 5 to 8 if given new incremental features.

IV. Experimental Analysis

In this section, we design a study using human subjects to demonstrate the effectiveness of the proposed coarse-to-fine feature selection method for SVD detection. Eight SVD patients (4 women and 4 men) and 12 healthy elderly subjects (6 women and 6 men) aged 62 to 76 were recruited for our experiments. The MRI characteristics of the 8 SVD patients are described in TABLE I. Each subject was asked to conduct all 17 motor actions in order. The execution time for each subject ranged from 4.35 minutes to 8.42 minutes.

TABLE I. The MRI characteristics of the SVD patients participating in our research

No.	Sex	Age	Fazekas scale (periventricular white matter)	Fazekas scale (deep white matter)	Number of lacunar infarcts
1	F	73	3	3	1
2	M	74	3	3	0
3	M	72	3	2	0
4	M	72	3	3	3
5	M	69	2	2	6
6	F	62	2	2	2
7	F	68	3	2	0
8	F	66	3	3	4

A feature selection experiment was then performed on the data of the 20 subjects using IRFFS. In our experiment, the training rate is set as 70%, namely $N_r = \lfloor 12 \times 70\% \rfloor + \lfloor 8 \times 70\% \rfloor = 13$, and the sampling times S is set as 1000. Considering there are three types of features, we set the minimum number of features $n_{\min}$ to 3, which makes it possible that the final selected feature set includes at least one feature from each type of features. Fig. 4 shows the average classification accuracy of the built random forest model at each iteration using the IRFFS method. As can be observed, for all numbers of trees B, the classification performance keeps growing at each iteration until the number of selected features is less than $n_{\min}$, which demonstrates that the proposed IRFFS method is effective in selecting significant features while dropping the redundant ones. As shown in TABLE II, the classification accuracy is

improved from 55.17% – 56.97% to 79.26% – 79.63% when the number of features is reduced from 739 to 5 after 8 iterations. Since the classification performance for the setting with B =100 is optimal, we choose the feature selection results from this setting as the final one, and we then obtain the 5 most significant features. These features are shown in TABLE III. Through observation, we find that these most discriminant features include three agility related features as well as two gait related features. This finding indicates that agility related features are helpful for SVD detection, and that fusing gait and agility related motor pattern features may have great potential for building a high-accuracy detection model.

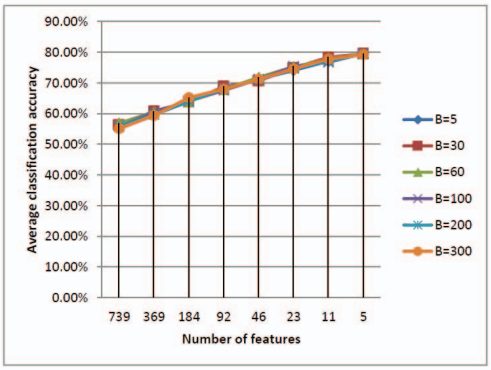

Fig. 4. The average classification accuracy of the random forest model at each iteration using the IRFFS method.

TABLE II. THE AVERAGE CLASSIFICATION ACCURACY OF THE RANDOM FOREST MODEL AT EACH ITERATION USING THE IRFFS METHOD

Number of Features	*B= 5*	*B=30*	*B=60*	*B=100*	*B=200*	*B=300*
739	56.39%	56.34%	**56.97%**	55.84%	55.90%	**55.17%**
369	59.79%	60.84%	60.61%	60.60%	59.71%	59.43%
184	64.30%	63.97%	64.07%	64.11%	64.00%	65.27%
92	68.10%	68.84%	67.93%	67.69%	68.27%	68.07%
46	71.27%	70.76%	71.90%	71.27%	71.11%	71.30%
23	74.17%	74.97%	75.34%	75.46%	74.76%	74.67%
11	77.14%	78.39%	77.67%	76.80%	77.17%	77.93%
5	79.60%	79.56%	79.61%	**79.63%**	79.51%	**79.26%**

We then compare the effectiveness of the IRFFS method with 7 other feature selection methods provided in Weka [4] through a classification test using both SVM [24] and ELM. As was the case for the feature selection experiment, we randomly select 70% of the samples for training and 30% of the samples for testing and repeat this sampling 1000 times. We use the LIBSVM tools [25] for SVM classification. For ELM, we use the implementation from [21]. In our experiments, the cost parameter of SVM is set as 3 and other parameters are set as the default ones. For ELM, the number of hidden nodes L is set as 1000, while the activation function g(x) is set as the sigmoid function. As shown in TABLE IV, the SVD detection accuracy of both SVM and ELM based on five features selected by the IRFFS method is

TABLE III. THE FIVE MOST DISCRIMINANT FEATURES FOR SVD DETECTION, SELECTED BY THE IRFFS METHOD WITH B=100

Feature type	Motor action	Feature extracted
Agility related feature	Finger nose test	Standard deviation of the moving velocity while pointing the nose using the right forefinger
Agility related feature	Tapping heel on the ground in rapid succession	Mean of the tapping velocity while picking up the entire right leg
Agility related feature	Rapid alternating movements of the hand	Mean of the pronation velocity of the right hand
Gait related feature	3-meter walk test and turning 180°	Mean of the sagittal-angular excursions of the left hip at right foot contact
Gait related feature	3-meter walk test and turning 180°	Standard deviation of the coronal-angular excursions of the left hip at right foot contact

TABLE IV. THE SVD DETECTION PERFORMANCES OF SVM AND ELM BASED ON FIVE FEATURES SELECTED BY DIFFERENT FEATURE SELECTION METHODS

Feature selection method	SVM	ELM
CfsSubsetEval	85.73%	86.85%
CorrelationAttributeEval	90.67%	90.66%
GainRatioAttributeEval	87.79%	88.60%
InfoGainAttributeEval	86.74%	85.56%
OneRAttributeEval	87.36%	88.65%
ReliefFAttributeEval	89.01%	88.33%
SymmetricalUncertAttributeEval	86.74%	85.49%
IRFFS(B =100)	**90.79%**	**91.43%**

better than that based on five features selected by the other feature selection methods, which demonstrates that the IRFFS method is more efficient in selecting the most significant features for accurate SVD detection from small-sampling data with high-dimensionality features.

After that, we check the discriminant ability of each kind of selected features using FIELM. Since there are two kinds of features, we train FIELM in two feature incremental types: 1) add agility related features on top of gait related features, and 2) add gait related features on top of agility related features. TABLE V shows the SVD detection performances of FIELM. As can be seen, both the selected gait related features and agility related features are significant for distinguishing SVD patients from healthy elderly persons because the average classification accuracy of 1000 tests using only gait related features and agility related features can reach 84.34% and 80.64% respectively. The results also indicate that the fine motor pattern features of upper and lower limbs and gait pattern features have different importance for high-accuracy SVD detection. Specifically, no healthy elderly person is wrongly classified as an SVD patient using the gait related features (the average false positive (FP) rate of 1000 tests based on gait related features is 0%). On the other hand, agility related features are more effective in reducing the probability of misclassifying SVD patients as healthy elderly persons, since the average true positive (TP) rate of 1000 tests using agility related features is higher than that based on gait related features. Furthermore, it can be noticed that the average classification accuracy increases by 7.1% and 10.78% when adding the

agility related features and the gait related features respectively, which demonstrates for the first time that fusing gait and agility related motor pattern features is helpful for building a more accurate SVD detection model. In the first feature incremental type, the average classification accuracy of 1000 tests using 5 fusion features reaches 91.44%. This experimental result verifies that through combining gait and agility related features, SVD can be accurately detected.

TABLE V. THE SVD DETECTION PERFORMANCE OF FIELM BASED ON FEATURES SELECTED BY THE IRFFS METHOD WITH B=100

Feature incremental types		Classification accuracy	TP rate	FP rate
Add agility related features on top of gait related features	Gait related features	84.34%	63.45%	0.00%
	Gait related features +Agility related features	91.44%	87.29%	5.45%
Add gait related features on top of agility related features	Agility related features	80.64%	65.83%	8.25%
	Agility related features +Gait related features	91.42%	87.27%	5.48%

V. CONCLUSION

This work presents an experimental investigation on significant motor pattern features for accurate detection of SVD. An effective coarse-to-fine feature selection method, which combines the IRFFS method and the FIELM model, is put forward to discover the most significant features from a high dimensional feature set of small-sampling data. Experimental results demonstrate that the proposed method performs well in selecting discriminant features for high-accuracy SVD detection. Furthermore, it also verifies that on top of gait and balance related features, agility related features are also useful to build a high-quality SVD classification model. In the future, we plan to conduct experiments on a larger population to validate the reliability of our findings.

ACKNOWLEDGEMENT

This research is supported by the Natural Science Foundation of China (NSFC) under Grant No.61572471, No.61502456 and No.61572004, the International Science & Technology Cooperation Program of China under Grant No.2014DFG12750 and the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative.

REFERENCES

[1] F. L. Karlijn, G.W. N. Anouk, A.R.G. Rob, J.B.O. Lucas, W.M.U. Inge, G.N. David, P.Z. Marcel, L. Frank-Erik, "Diffusion Tensor Imaging and Gait in Elderly Persons With Celebral Small Vessel Disease", Stroke, vol. 42, pp. 373-379, 2011.

[2] A. Soumaré, A. Elbaz, Y.C. Zhu, P. Maillard, F. Crivello, B. Tavernier, C. Dufouil, B. Mazoyer, C. Tzourio, "White Matter Lesions Volume and Motor Performances in the Elderly", Ann Neurol, vol. 65, pp. 706-715, 2009.

[3] L. Pantoni, "Celebral Small Vessel Disease: From Pathogenesis and Clinical Characteristics to Therapeutic Challenges", Lancet Neurol, vol. 9, 689-701, July 2010.

[4] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, "The WEKA Data Mining Software: An Update", SIGKDD Explorations, vol 11(1) , 2009.

[5] M. Termenon, M. Graña, A. Savio, et al, "Brain MRI morphological patterns extraction tool based on Extreme Learning Machine and majority vote classification". Neurocomputing, 174: 344-351, 2016.

[6] G. B. Huang, Q. Y. Zhu, C.K. Siew, "Extreme Learning Machine: a New Learning Scheme of Feedfoward Neural Networks", Neurocomputing, vol. 70, pp. 489-501, 2006.

[7] L. S. Hu, Y. Q. Chen, S. Q. Wang, Z. Y. Chen, "b-COELM: A fast, lightweight and accurate activity recognition model for mini-wearable devices", Pervasive and Mobile Computing, vol. 15, pp. 200-214, 2014.

[8] Y. Q. Chen, H. C. Yu, C. Y. Miao, et al., "Using motor patterns for stroke detection", Science (Supplement), 12-15, 2015.

[9] X. L. Jiang, J. F. Liu, Y. Q. Chen, D. J. Liu, Y. Gu, Z. Y. Chen, "Feature Adaptive Online Sequential Extreme Learning Machine for lifelong indoor localization". Neural Computing and Applications, 27(1): 215-225,2016.

[10] Y. Gu, J. F. Liu, Y. Q. Chen, X. L. Jiang, "Constraint Online Sequential Extreme Learning Machine for lifelong indoor localization system", IJCNN, pp. 732-738, 2014.

[11] J. Shotton, A.W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, "Real-Time Human Pose Recognition in Parts from Single Depth Images", in Proceedings of the IEEE Conference on Computer Vision and Patterns Recognition, Colorado, USA, pp.1297-1304, 2011.

[12] J. C. M. Zijlmans, P. J. E. Poels, J. Duysens, J. Straaten, T. Thien, M.A. Hof, H.O.M. Thijssen, M.W.I.M. Horstink, "Quantitative Gait Analysis in Patients with Vascular Parkinsonism", MovDisord, vol. 11, pp. 501-508, 1996.

[13] L. Breiman, "Random Forests", Machine Learning, vol. 45, pp. 5-32, October 2001.

[14] J. R. Quinlan, "Induction of Decision Trees",. Machine Learning, vol.1, pp. 81-106, 1986.

[15] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, "Classification and regression trees", Monterey, 1984.

[16] L. Rokach, O. Maimon, "Top-down induction of decision trees classifiers-a survey", IEEE Transactions on Systems, Man, and Cybernetics, Part C vol. 35 (4), pp. 476-487, 2005.

[17] T. K. Ho. "A Data Complexity Analysis of Comparative Advantages of Decision Forest Constructors", Pattern Analysis and Applications, pp. 102-112, 2002.

[18] K. Han, D. Yu, I. Tashev, "Speech Emotion Recognition Using Deep Neural Network and Extreme Learning Machine", in Proceedings of the 15th Annual Conference of the International Speech Communication Association, MAX Atria, Singapore, pp.14-18, 2014.

[19] G. B. Huang, L. Chen, and C. K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," IEEE Trans. Neural Netw., vol. 17, no. 4, pp. 879–892, Jul. 2006.

[20] G. B. Huang and L. Chen, "Convex incremental extreme learning machine,"Neurocomputing, vol. 70, no. 16–18, pp. 3056-3062, Oct. 2007.

[21] G. B. Huang, et al., "Extreme learning machine for regression and multiclass classification". Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 42, no. 2, pp. 513-529, 2012.

[22] R. Chandra, R. Mahajan, T. Moscibroda, "A Case for Adapting Channel Width in Wireless Networks", In: Proceedings of the ACM SIGCOMM 2008 conference on Data communication. vol. 38, pp. 135-146, 2008.

[23] C. R. Rao and S. K. Mitra, "Generalized Inverse of Matrices and Its Applications". New York: Wiley, 1971.

[24] C. J. C. BURGES, "A tutorial on support vector machines for pattern recognition", Data mining and knowledge discovery, 2(2): 121-167, 1998.

[25] C. C. Chang and C. J. Lin, "LIBSVM : a library for support vector machines", 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.