



A Novel Functional Link Network Stacking Ensemble with Fractal Features for Multichannel Fall Detection

Ahsen Tahir^{1,2} · Gordon Morison¹ · Dawn A. Skelton³ · Ryan M. Gibson¹

Received: 28 March 2020 / Accepted: 10 June 2020 / Published online: 29 July 2020
© The Author(s) 2020

Abstract

Falls are a major health concern and result in high morbidity and mortality rates in older adults with high costs to health services. Automatic fall classification and detection systems can provide early detection of falls and timely medical aid. This paper proposes a novel Random Vector Functional Link (RVFL) stacking ensemble classifier with fractal features for classification of falls. The fractal Hurst exponent is used as a representative of fractal dimensionality for capturing irregularity of accelerometer signals for falls and other activities of daily life. The generalised Hurst exponents along with wavelet transform coefficients are leveraged as input feature space for a novel stacking ensemble of RVFLs composed with an RVFL neural network meta-learner. Novel fast selection criteria are presented for base classifiers founded on the proposed diversity indicator, obtained from the overall performance values during the training phase. The proposed features and the stacking ensemble provide the highest classification accuracy of 95.71% compared with other machine learning techniques, such as Random Forest (RF), Artificial Neural Network (ANN) and Support Vector Machine. The proposed ensemble classifier is $2.3\times$ faster than a single Decision Tree and achieves the highest speedup in training time of $317.7\times$ and $198.56\times$ compared with a highly optimised ANN and RF ensemble, respectively. The significant improvements in training times of the order of $100\times$ and high accuracy demonstrate that the proposed RVFL ensemble is a prime candidate for real-time, embedded wearable device-based fall detection systems.

Keywords Fall detection · Machine learning · Artificial neural network · RVFL neural network · Ensemble learning · Fractal features

Introduction

Falls are a major health hazard for older adults and result in high morality and injury rates [55]. A large percentage

of fall incidents, up to 62%, result in immobility [9, 34]. Falls result in high costs to the national health service [55]. Early detection of falls and immediate medical aid can save lives and reduce death by 80% [39]. Fall Detection Systems (FDSs) play an important role in timely medical aid provision through early detection of falls [39]. FDS can be sensor based [38, 41, 64] or camera based [15, 22, 30, 37]. Sensor-based systems can be wearable [26, 41] and smartphone-based [21, 46] accelerometers or gyroscopes, while environmental sensors frequently use infrared [7, 14], pressure sensors [59] and WiFi-based sensing devices [17, 54], which utilise fluctuations in channel state information amplitude at the WiFi receiver to sense activities. The readings from these sensors are used to detect and classify falls from Activities of Daily Life (ADL).

The sensor signals are evaluated with signal processing algorithms to extract features for classification. Machine learning and neural network algorithms are then frequently used to classify and detect falls from the extracted features by a processing device [21, 26, 62]. Machine learning techniques and neural network models have a significant focus on randomised algorithms in recent years, due to

✉ Ahsen Tahir
ahsen.tahir@gcu.ac.uk

Gordon Morison
gordon.morison@gcu.ac.uk

Dawn A. Skelton
dawn.skelton@gcu.ac.uk

Ryan M. Gibson
ryan.gibson@gcu.ac.uk

¹ School of Computing, Engineering and Built Environment, Glasgow Caledonian University, Glasgow G4 0BA, UK

² Department of Electrical Engineering, University of Engineering and Technology, Lahore 54890, Pakistan

³ School of Health and Life Sciences, Glasgow Caledonian University, Glasgow G4 0BA, UK

their asymptotically faster runtimes and computationally efficient models [48, 56]. The main idea behind utilising randomised learning for neural networks is to assign random weights and biases to neural network inputs and compute output parameters by solving a linear system [57]. Random Vector Functional Link (RVFL) neural networks introduced by Pao et al. in [42] utilise randomness for a subset of weights and biases between the input and a single hidden layer, which are kept fixed during the training procedure. Unlike single hidden layer Multi-Layer Perceptron (MLP), RVFL networks have direct links between the inputs and the output. The output weights can be computed from a closed form least-squares method. RVFL networks are computationally efficient and fast learners, unlike traditional neural networks [43]. Therefore, RVFL network is a prime candidate for fast ensemble techniques.

Fractal dynamics is an essential part of complex non-linear dynamic systems which are chaotic in nature and appears in state space representations of such systems with time-evolving trajectories [35]. The human movements manifest as a result of complex non-linear interactions representative of a complex non-linear dynamic system [45] and can be analysed as a chaotic system exhibiting fractal dynamics, while the current work in non-linear dynamics is limited to the analysis of human movements during walking [52]. Other activities such as falls can be analysed with fractal dynamics. Real-world fractals are statistically self-similar patterns and signals, where the whole is statistically similar to its components. The generalised Hurst exponent is related to fractal characteristics of a signal and is used for fractal analysis of time-varying biomedical signals [44]. The fractal characteristics of a signal have a positive correlation with its irregularity, according to Mandelbrot [33] and the generalised Hurst exponent can be used as a measure of signal irregularity.

We utilise RVFL neural networks as base classifiers for our proposed ensemble method. Each base classifier is trained in a highly optimised feature space to achieve high classification accuracy. We propose fractal feature-based classification of falls, along with Discrete Wavelet Transform (DWT) coefficient features. Hurst exponent values are used as fractal features to represent the accelerometer signals for falls and ADL. Fall signals traditionally consist of a frequent single-magnitude spike, as opposed to continuous and lower magnitude variations for other activities [23] and potentially have different irregularity characteristics, which can potentially be exploited as features for classification. The Hurst exponents are calculated from the Signal Summation Conversion (SSC) method [12]. The DWT multilevel wavelet transform is performed in parallel with the SSC computations and the resulting DWT coefficients and

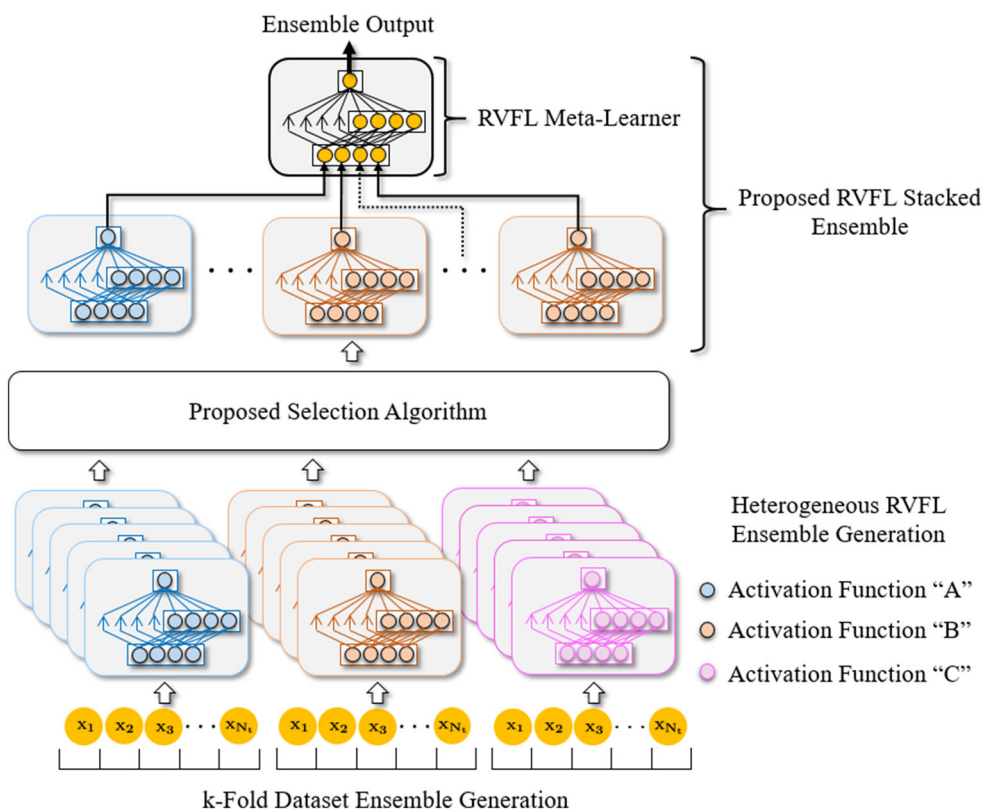
Hurst exponents are used to train various classifiers for fall detection.

Furthermore, our work proposes a novel ensemble of RVFL neural networks combined by an RVFL network meta-learner as illustrated in Fig. 1 for the final classification of falls from ADLs, to achieve low latency and fast training for the ensemble learner. The base RVFL classifier response can be potentially diverse due to a random subset of parameters used in RVFL networks. The proposed technique introduces heterogeneity by choosing from a set of different kernel functions for the network and selecting classifiers from different folds of the same k -fold training procedure. This encourages generation of a larger number of base classifiers within the same training procedure for lower runtime costs. However, the speed advantage of the RVFL ensemble can be lost if an efficient procedure for selection of RVFL base classifiers is not adopted. Additionally, we also propose an efficient scheme for selection of individual classifiers.

Our work determines diversity of models from overall performance measures of base classifiers, namely total True Positives (TP) and True Negatives (TN). The insight is based on the observation that two models with similar accuracies can be inherently different due to the way they classify positives and negatives. A model with high TP and low TN values may have a similar accuracy to a model with low TP and high TN values, since accuracy depends on the aggregate sum of the two counts. However, the two models are inherently diverse. The technique determines a diversity indicator from overall performance measures (TP and TN values) computed while training the model, instead of finding diversity amongst the models from individual inputs and their corresponding classification outputs. Calculating diversity from each input space sample is not computationally feasible for comparing a large number of models. An Aggregate Performance as Diversity Indicator (APDI) is constructed from their difference and the concept is applied to models with same or different accuracies. The proposed RVFL ensemble and selection algorithm-based FDS is illustrated in Fig. 1, while an overview of the fall detection process is shown in Fig. 2. As illustrated, the accelerometer signals from a wearable sensor device are transmitted to a local processing system through the WiFi Router for fall classification. On detection of a fall event, the nearest medical aid centre is notified for timely medical assistance. The contributions of our work are summarised as follows:

- We proposed the use of the generalised Hurst exponent for fall classification as a metric to characterise the irregularity of a signal. Mandelbrot in [33] demonstrated that fractal dimensions' increase with the irregularity of a one-dimensional curve and have a positive

Fig. 1 RVFL stacked ensemble

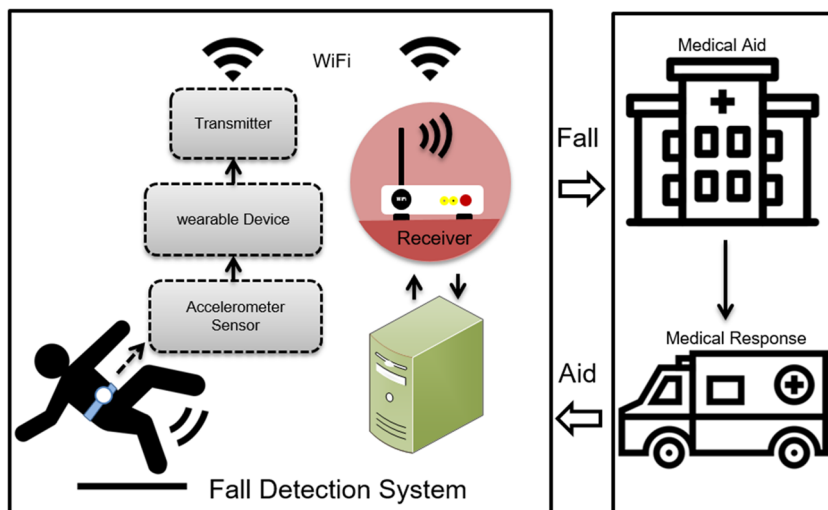


correlation with signal irregularity. The generalised Hurst exponent is related to the fractal dimensions of a signal and is used to determine the fractal dimension of one-dimensional signals [44]. The generalised Hurst exponent is therefore leveraged as a discriminating feature for representing the irregularity characteristics of a signal to train machine learning algorithms for fall classification. The technique is based on the observation that fall accelerometer signals consist of a spike and have different irregularity characteristics than ADL

signals, which possess higher irregularity. Our fractal features demonstrate high accuracy results for traditional machine learning algorithms including 88.57% for Support Vector Machine (SVM) and 90% for both *K*-Nearest Neighbour (KNN) and RF. We demonstrate randomised neural networks on proposed features with a single RVFL neural network and achieve the highest accuracy of 91.43%.

- We further improved the classification model and proposed a unique and novel stacking RVFL ensemble

Fig. 2 Fall detection system overview



with an RVFL meta-learner for fall detection and classification with the highest classification accuracy of 95.71%. To the best of our knowledge, this is the first instance of RVFL stacking ensembles combined with a novel selection algorithm for classification tasks and detection of falls from ADL. Our stacking ensemble achieves significant performance in training speed in comparison with non-ensemble and ensemble learning techniques, while maintaining the highest accuracy. The proposed ensemble classifier is $2.3\times$ faster than a single Decision Tree (DT), while achieving even higher speedups in training times of $317.7\times$ and $198.56\times$ compared with highly optimised ANN and RF ensemble, respectively.

- We also proposed a novel and fast selection method for base classifiers determined by a diversity indicator obtained from overall performance values during training. We utilise the variations in TP and TN values amongst base classifiers to indicate diversity between classifiers. The diversity criteria are based on the insight obtained through experimental results and observations that two RVFL base models with similar accuracy can be inherently diverse based on whether the accuracy performance has been achieved through higher TP or higher TN values. To the best of our knowledge, this is the first instance of the difference in TP and TN values being utilised to determine diversity for base classifiers. We design and develop a heuristic for diversity indication, Aggregate Performance as Diversity Indicator (APDI) for selection of our base models based on our proposed diversity criteria through heuristic-based search algorithm. The base classifier selection method based on the aggregate performance values (TP and TN) not only results in fast selection during training but also achieves high accuracy with a small number of base models.

The next “[Related Work](#)” section discusses the related work and “[Mathematical Techniques](#)” section explains the mathematical techniques used. The proposed algorithm with the ensemble technique is discussed in “[Proposed Algorithm](#)”. Section “[Methodology](#)” presents methodology and “[Results and Discussion](#)” section provides results and discussion.

Related Work

There has been a number of recent research publications on detecting fall events based on wearable sensors using traditional machine learning techniques. Hsieh et al. [19] proposed a fall detection algorithm that utilises both machine learning and threshold-based techniques for detection of falls from accelerometer signals with

high accuracy, above 98%. Sukor et al. [51] leveraged time and frequency space features including energy and power spectrum of accelerometer signals for fall detection. Principal Component Analysis (PCA) was performed on feature space to select the principal components and various machine learning classifiers including DT and SVM were used for fall detection. Ramon et al. [47] utilised a multiple-sensor body area network with a smartphone for sensing, processing and classification of falls and ADLs. A number of classifiers including SVM, KNN, Naive Bayes and DT were applied and analysis of variance algorithm was used for validation of different algorithms.

Ensemble techniques have also been utilised for fall detection. The latest work by Chelli et al. [5] utilises Ensemble Bagged Trees (EBT) for classification of falls with a number of time and frequency features with an accuracy of 97.7%. In [58], authors utilise convolutional layers to extract features from images and a bagged tree ensemble is then used for fall classification. Nguyen et al. [36] proposed RF for detection of falls with a number of time features including signal energy and achieved an accuracy of 94.37% for fall detection. However, generation of a large number of trees is computationally expensive and we show that our proposed method is faster than a single DT. Yang et al. [63] combined tree classifiers through a diversity-based technique for RF ensemble using weights for each sample. The tree weights are learned through convex quadratic programming. However, our method utilises RVFL neural networks for fast training speed and is $2.3\times$ faster than a single decision tree. Moreover, our proposed method uses a simple selection procedure based on aggregate performance metrics obtained from each model which improves the overall accuracy and reduces the selection time cost.

Ensemble techniques for combining neural networks were also leveraged for fall classification. Recently, Chen et al. [6] proposed ensemble of stacked AutoEncoders (AE) along with One-Class Classification based on the Convex Hull (OCCCH) technique. Furthermore, the authors utilised two stage ensemble method with majority voting in the first and weighted ensemble in the second stage. Khan et al. [27] proposed an AE ensemble for processing accelerometer and gyroscope signals for classification of falls. Furthermore, the authors utilise a majority voting scheme to combine the results of the classifiers. Wen et al. [60] presented an ensemble of CNN, where each CNN outputs a probability for each class. The probabilities are then combined using a probability-based fusion method and the maximum probability is used for the final classification results. However, deep neural networks, such as AEs, CNNs, etc., are computationally extensive and suffer from high runtime costs associated with generation of base classifiers for deep neural networks. Our proposed technique offers a fast RVFL

ensemble technique with RVFL meta-learner to combine the outputs, along with an aggregate performance-based diversity indicator for selection of base learners, resulting in high accuracy and low runtime costs.

Randomised algorithms [32] have received a significant focus in recent years for large-scale computing applications, due to their asymptotically faster runtimes and efficient numerical implementations. Neural networks and machine learning models have also exploited randomised algorithms for faster training [48, 56]. To the best of our knowledge, this is the first instance of randomised weights-based RVFL neural networks for fall detection. RVFL neural networks use a subset of randomised weights/biases and were proposed by Pao et al. in [42], while generalisation ability and learning characteristics were discussed in [43]. Zhou et al. [67] presented an online version of RVFL with sequential learning for modelling of dynamic time-varying complex systems. The proposed method is applied to the prediction of quality indices for an industrial furnace process. Xu et al. used the RVFL networks for learning of spatio-temporal processes [61]. Maeda et al. [31] used a convolutional coding-based deep RVFL neural network for distress classification of roads. Tian et al. [53] used RVFL networks for recognition of intrusion signal in an optical fiber warning system. Cecotti et al. [4] used deep RVFL neural networks for recognition of handwritten characters. Scardapane et al. [49] present Bayesian inference techniques for data modelling with RVFL networks, while Dai et al. [11] used RVFL networks for the diagnosis of Alzheimer's disease. The authors also used their model to determine the progression of disease. Katuwal et al. in [25] proposed an ensemble of RVFL neural network with DTs. RVFL network is used for initial division of data into classes and DTs are applied to the obtained classes of data for final classification. However, the DTs have a higher runtime cost and take away the speed advantage of RVFL networks. In our work, the proposed RVFL ensemble is $2.3\times$ faster than a single DT. Furthermore, Katuwal et al. in [24] also proposed an ensemble of deep RVFL network, which utilises all the hidden layers of a single deep network to obtain a separate output from each layer and calculates an ensemble output based on average or a majority vote. However, deep RVFL ensembles have high computational complexity and do not provide the speed advantage of our proposed RVFL ensemble.

Fractal features have also been used in biomedical systems with machine learning techniques but are limited to diagnosis of anomalies or have been used only for human gait analysis. Fractal dynamics of walk and human gait have been analysed in [18, 52] and [50]. Various anomalies have been detected in biological systems by the use of fractal dimensions [28, 66]. Koutsiana et al. [28] detected fetal heart sounds by computing fractal dimensions of wavelet

transformed signals. Anomalies in the human brain have also been detected by Zhang et al. in [66] by utilising fractal dimensions. However, to the best of our knowledge, the generalised Hurst exponent has not been used for activities in general and falls in particular as an irregularity measure of the obtained signals.

Mathematical Techniques

Discrete Wavelet Transform

The accelerometer signals representing the tri-axis of motion along x , y and z axes can be mathematically represented as $\mathbf{a}_x = \{a_x(n)\}$, $\mathbf{a}_y = \{a_y(n)\}$ and $\mathbf{a}_z = \{a_z(n)\}$, where $n = \{1, \dots, N\}$ and $N = 128$ samples for the evaluated window size. The tri-axis accelerometer signals \mathbf{a}_x , \mathbf{a}_y and \mathbf{a}_z are illustrated in Fig. 3 in red, orange and blue, respectively. The DWT of tri-axis accelerometer signals is a projection of each tri-axis accelerometer signal on a family of wavelet basis functions $\phi_{i,k}(n)$ and $\psi_{i,k}(n)$. The basis functions are obtained from dilations and translations of the mother wavelet $\psi(n)$ and scaling function $\phi(n)$ and are as follows:

$$\phi_{i,k}(n) = 2^{-\frac{i}{2}} \phi(2^{-i}n - k) \quad (1)$$

$$\psi_{i,k}(n) = 2^{-\frac{i}{2}} \psi(2^{-i}n - k) \quad (2)$$

where k are discrete translations and 2^i represent dyadic dilations. The DWT coefficients of each tri-axis accelerometer signal, generally represented as $a_{dim}(n)$, where dim represents the dimensions of motion along x , y or z axis, can be given as:

$$\mathcal{A}_{i,dim}(k) = \sum_n a_{dim}(n) \phi_{i,k}(n) \quad (3)$$

$$\mathcal{D}_{i,dim}(k) = \sum_n a_{dim}(n) \psi_{i,k}(n) \quad (4)$$

where $\mathcal{A}_{i,dim}$ represents wavelet approximation coefficient vector and $\mathcal{D}_{i,dim}$ represents wavelet detail coefficient vector for each of the three axes of motion where dim can be x , y or z , while k is the shifting index of mother and scaling wavelet functions.

The low pass wavelet coefficients are also known as approximations. The approximations $\mathcal{A}_{1,dim}$ are further used as input signals in Eqs. 3 and 4 to generate level-2 DWT approximations $\mathcal{A}_{2,dim}$ and details $\mathcal{D}_{2,dim}$. The level-2 DWT approximations $\mathcal{A}_{2,dim}$ are then used as input signals in Eqs. 3 and 4 to generate level-3 DWT approximations $\mathcal{A}_{3,dim}$ and details $\mathcal{D}_{3,dim}$. Similarly, level-3 DWT approximations $\mathcal{A}_{3,dim}$ are used to generate the final level-4 DWT approximations $\mathcal{A}_{4,dim}$ and details $\mathcal{D}_{4,dim}$.

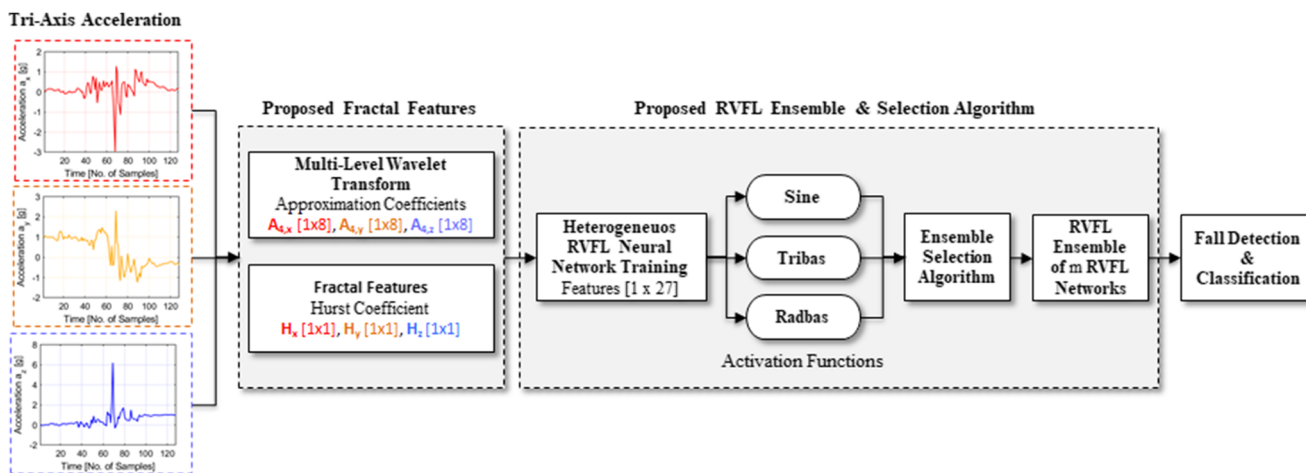


Fig. 3 Proposed fall detection system

The final level-4 DWT approximations $\mathcal{A}_{4,dim}$ are a vector of size $[1 \times 8]$ for each of the axes, where dim can be x , y or z . The 4-level DWT approximations $\mathcal{A}_{4,x}$, $\mathcal{A}_{4,y}$ and $\mathcal{A}_{4,z}$ for each of the axis x , y or z are directly used as features for classification in the FDS system, along with the fractal features as illustrated in Fig. 3. The next section discusses the SSC method for computation of fractal features of accelerometer signals of falls and activities.

Signal Summation Conversion

The signal summation method for computation of Hurst exponent [12] is also implemented on each of the tri-axis accelerometer signals $\mathbf{a}_x = \{a_x(n)\}$, $\mathbf{a}_y = \{a_y(n)\}$, $\mathbf{a}_z = \{a_z(n)\}$, where $n = \{1, \dots, N\}$ and $N = 128$ samples along the three axes of motion x , y and z . The steps of the process on each tri-axis accelerometer signal, generally represented as $a_{dim}(n)$, where dim denotes one of the three dimensions or axes of motion x , y or z , can be given as:

1. Compute the cumulative sum signal $s_{dim}(n)$ of the accelerometer signal $a_{dim}(n)$ as follows:

$$s_{dim}(n) = \sum_{n=1}^N a_{dim}(n) \tag{5}$$

2. Partition N samples of the cumulative sum signal $s_{dim}(n)$ for each of the three accelerometer axes $dim = \{x, y, z\}$, into N/w non-overlapping windows of size $w = \{2, 4, \dots, N/2, N\}$.
3. Perform the detrending process on the signal obtained from the previous step. In our work, we perform the bridge detrending process [3], which involves computing a separate line equation that connects the first and last points in the window for each partition. Given a window size w and window partition index $j = \{1, 2, \dots, N/w\}$, the index of the first and last points in

the j^{th} window partition can be given as $jw - w + 1$ and jw . The slope $sl_{dim,j}$ and signal magnitude axis intercepts $b_{dim,j}$ for each window partition j , for each of the three accelerometer axes x , y and z respectively, are computed as:

$$sl_{dim,j} = \frac{s_{dim}(jw - w + 1) - s_{dim}(jw)}{w - 1} \tag{6}$$

$$b_{dim,j} = s_{dim}(jw - w + 1) - sl_{dim,j} \tag{7}$$

The function $d_{dim,j}$ representing the line equation for each accelerometer axis and each window partition j can be given as:

$$d_{dim,j}(k_w) = sl_{dim,j}k_w + b_{dim,j} \tag{8}$$

Each equation is then subtracted from the signal in the respective partition. The detrended signal $\acute{s}_{dim,j}$ for each partition j is given as:

$$\acute{s}_{dim,j}(wj - w + k_w) = s_{dim}(wj - w + k_w) - d_{dim,j}(k_w) \tag{9}$$

where $k_w \in \{1, \dots, w\}$ is the signal index in each window partition j , relative to its start.

4. Compute the mean $\bar{s}_{dim,j}$ and standard deviations $\sigma_{dim,j}$, for each of the three accelerometer axes respectively, for each detrended window j from the following equations:

$$\bar{s}_{dim,j} = \frac{1}{w} \sum_{k_w=1}^w \acute{s}_{dim,j}(wj - w + k_w) \tag{10}$$

$$\sigma_{dim,j} = \sqrt{\frac{1}{w - 1} \sum_{k_w=1}^w \left(\acute{s}_{dim,j}(wj - w + k_w) - \bar{s}_{dim,j} \right)^2} \tag{11}$$

where $j = \{1, 2, \dots, N/w\}$ each for a given window size w .

5. Compute the mean of standard deviations over all windows $j = \{1, 2, \dots, N/w\}$ for a given window size w , for each given value of $w = \{2, 4, 8 \dots, N\}$ as:

$$\bar{\sigma}_{dim,w} = \frac{1}{N/w} \sum_{j=1}^{N/w} \sigma_{dim,j} \tag{12}$$

6. The Hurst exponents H_{dim} , for the three accelerometer axes, $dim = \{x, y, z\}$, are related to the mean of standard deviation $\bar{\sigma}_{dim,w}$ for each window size w as given in Eq. 13, where ρ is the constant of proportionality. H_{dim} can be computed from the slope of the least squares regression lines $\log \bar{\sigma}_{dim,w}$ versus $\log w$ according to Eq. 14.

$$\bar{\sigma}_{dim,w} \propto \rho w^{H_{dim}} \tag{13}$$

$$H_{dim} = \lim_{w \rightarrow 0} \frac{\log \bar{\sigma}_{dim,w}}{\log w} \tag{14}$$

The Hurst exponents H_x , H_y and H_z represent fractal features for each dimension dim , with $dim = \{x, y, z\}$ and are used along with the 4th-level wavelet approximation coefficient vectors $\mathcal{A}_{4,x}$, $\mathcal{A}_{4,y}$ and $\mathcal{A}_{4,z}$ as input features for the RVFL neural network.

RVFL Neural Network

The RVFL neural network is a single hidden layer network, first introduced by Pao et al. in [42] and [43]. It is characterised by direct links between the input and the output layers, apart from the conventional connections between the input and hidden layers, as illustrated in Fig. 4. Hidden layer in an RVFL neural network is also known as the enhancement layer. The weights and biases between the input and enhancement layers are randomly initialised in an RVFL network and remain constant during the training phase, while the weights that connect the input and enhancement layer nodes to the output layer are learnt.

Given inputs $\mathbf{x}_{di} \in \mathbb{R}^r$ and target output $y_{di} \in \mathbb{R}$, where $di = \{1, \dots, N_t\}$ is the data index and $r = 27$ are the number of wavelet and fractal input features. The RVFL network will have r input neurons and 1 output neuron. Let G be the number of enhancement nodes and α_g denote random weights between the input and enhancement nodes, where $g = \{1, \dots, G\}$ represents the index of the enhancement nodes. Then, $\alpha_1 = [\alpha_{1,1} \dots \alpha_{1,r}]$ represents the random weights initialised between the first enhancement node $g = 1$ and all the input nodes, which are equal to the number of features r . The output of the g^{th}

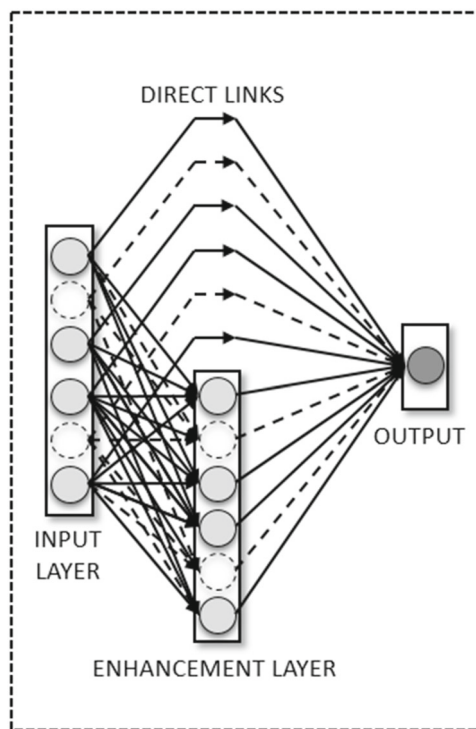


Fig. 4 RVFL Neural Network

enhancement node y_g for the di^{th} data input can be given as:

$$y_g(\mathbf{x}_{di}) = f_{act}(\alpha_g \cdot \mathbf{x}_{di}^T + b_g) \tag{15}$$

where, f_{act} represents an activation function for the neural network. In matrix form, the overall input matrix \mathbf{X} for the output node of RVFL network can be represented as a concatenation of two matrices \mathbf{X}_1 and \mathbf{X}_2 , where \mathbf{X}_1 represents inputs from the input layer and \mathbf{X}_2 represents inputs from the enhancement layer:

$$\mathbf{X} = [\mathbf{X}_1 \ \mathbf{X}_2], \text{ where} \tag{16}$$

$$\mathbf{X}_1 = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{N_t} \end{bmatrix} = \begin{bmatrix} x_{1,1} & \dots & x_{1,r} \\ \vdots & \ddots & \vdots \\ x_{N_t,1} & \dots & x_{N_t,r} \end{bmatrix}, \text{ and} \tag{17}$$

$$\mathbf{X}_2 = \begin{bmatrix} f_{act}(\alpha_1 \cdot \mathbf{x}_1^T + b_1) & \dots & f_{act}(\alpha_g \cdot \mathbf{x}_1^T + b_g) \\ \vdots & \ddots & \vdots \\ f_{act}(\alpha_1 \cdot \mathbf{x}_{N_t}^T + b_1) & \dots & f_{act}(\alpha_g \cdot \mathbf{x}_{N_t}^T + b_g) \end{bmatrix} \tag{18}$$

Given weights β of the direct links to the output node, both from the input and the enhancement nodes. The outputs \mathbf{t} of the RVFL network can be given as:

$$\mathbf{X}\beta = \mathbf{t}, \tag{19}$$

where \mathbf{t} is the target output vector,

$$\mathbf{t} = \begin{bmatrix} t_1 \\ \vdots \\ t_{N_t} \end{bmatrix}, \tag{20}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{r+G} \end{bmatrix} \tag{21}$$

From Eq. 19, the output weights $\boldsymbol{\beta}$ can be calculated directly by the Moore-Penrose method given in [20] or the ridge regression [1, 65], as presented in Eq. 22,

$$\begin{aligned} \boldsymbol{\beta} &= \mathbf{X}^\dagger \mathbf{t} \\ &= (\mathbf{X}^T \mathbf{X} + \frac{\mathbf{I}}{C_t})^{-1} \mathbf{X}^T \mathbf{t} \end{aligned} \tag{22}$$

Proposed Algorithm

The proposed algorithm takes tri-axis accelerometer values for all the three axes of motion. The signals are divided into window segments of size 128 and processed to extract wavelet low pass coefficients and fractal features. The selected Daubechies 4 wavelet coefficients and generalised Hurst exponents computed for all the three axes of motion are used to train RVFL models. The selected RVFL models are then used to generate an RVFL ensemble of RVFL models. The proposed algorithm is divided into two sections feature processing and RVFL Ensemble classifier, which are described next and illustrated in Figs. 5 and 6, respectively.

Feature Processing

1. Divide each tri-axis accelerometer signal, generally represented as \mathbf{a}_{dim} , where dim denotes one of the three dimensions of motion x , y or z , into window segments of size $N = 128$ samples.
2. Compute level-4 DWT approximation coefficients $\mathcal{A}_{4,\text{dim}}$ for the accelerometer signal \mathbf{a}_{dim} for each of the three dimensions of motion, with dim equal to x , y or z axis.
 - (a) Perform zero padding for each of the tri-axis accelerometer signal \mathbf{a}_{dim} and compute convolution $\mathbf{a}_{\text{dim}} \otimes \mathbf{h}_{\text{db4}}$ of each signal with Daubechies 4 wavelet filter coefficients \mathbf{h}_{db4} and down sample by 2 to find level-1 approximation coefficients $\mathcal{A}_{1,\text{dim}}$ for each of the axis x , y and z .
 - (b) Zero pad level-1 wavelet approximation coefficients $\mathcal{A}_{1,\text{dim}}$ and compute convolution $\mathcal{A}_{1,\text{dim}} \otimes$

\mathbf{h}_{db4} for each of the three axes with Daubechies 4 wavelet filter coefficients \mathbf{h}_{db4} and down sample by 2 to find level-2 approximation coefficients $\mathcal{A}_{2,\text{dim}}$.

- (c) Zero pad level-2 wavelet approximation coefficients $\mathcal{A}_{2,\text{dim}}$ and compute convolution $\mathcal{A}_{2,\text{dim}} \otimes \mathbf{h}_{\text{db4}}$ for each of the three axes with Daubechies 4 wavelet filter coefficients \mathbf{h}_{db4} and down sample by 2 to find level-3 approximation coefficients $\mathcal{A}_{3,\text{dim}}$ for each of the three dimensions of motion x , y and z .
 - (d) Similarly, perform zero padding of level-3 approximation coefficients $\mathcal{A}_{3,\text{dim}}$ and compute convolution $\mathcal{A}_{3,\text{dim}} \otimes \mathbf{h}_{\text{db4}}$ for each of the three axes of motion with Daubechies 4 wavelet filter coefficients \mathbf{h}_{db4} , which are then down sampled by 2 to find level-4 approximation coefficients. The level-4 DWT coefficients $\mathcal{A}_{4,\text{dim}}$ for each dimension of motion are used as features in the training and testing of RVFL neural networks for generation of an RVFL stacking ensemble of RVFL networks.
3. Compute the generalised Hurst exponent H_{dim} with the SSC method to determine and represent the irregularity characteristics of falls and activities, where H_{dim} represents each of the three axes generalised Hurst exponents H_x , H_y and H_z for each of the three accelerometer signals \mathbf{a}_x , \mathbf{a}_y and \mathbf{a}_z .
 - (a) Compute the cumulative sum \mathbf{s}_{dim} of accelerometer signal \mathbf{a}_{dim} , for each of the three axes.
 - (b) Divide the signal \mathbf{s}_{dim} into N/w non overlapping windows of size $w = \{2, \dots, N/2, N\}$.
 - (c) Compute bridge detrending [3], using vector $\mathbf{d}_{\text{dim},j}$ which represent lines from the first to the last point in each window j . Perform detrending for the tri-axis cumulative accelerometer signals for all j partitions, $\mathbf{s}_{\text{dim},j} - \mathbf{d}_{\text{dim},j}$.
 - (d) Compute the standard deviation $\sigma_{\text{dim},j}$ of all the three axis detrended signals $\mathbf{K}_{\text{dim},j}$ for each window partition j .
 - (e) Compute the mean of standard deviation for each axis $\bar{\sigma}_{\text{dim},w}$ and for each window size w .
 - (f) Compute the Hurst exponent H_{dim} , which represents H_x , H_y and H_z for all the three accelerometer signals from the slope $\frac{\log \bar{\sigma}_{\text{dim},w}}{\log w}$.
 4. Construct a feature set of $r = 27$ features with $[3 \times 8]$ Daubechies 4 wavelet coefficients consisting of, $[1 \times 8]\mathcal{A}_{4,x}$, $[1 \times 8]\mathcal{A}_{4,y}$, and $[1 \times 8]\mathcal{A}_{4,z}$ coefficients along the three axes of motion and $[1 \times 3]$ generalised Hurst exponent features, $\{H_x, H_y, H_z\}$ from the tri-axis accelerometer values for each fall and activity signal, as illustrated in Fig. 5.

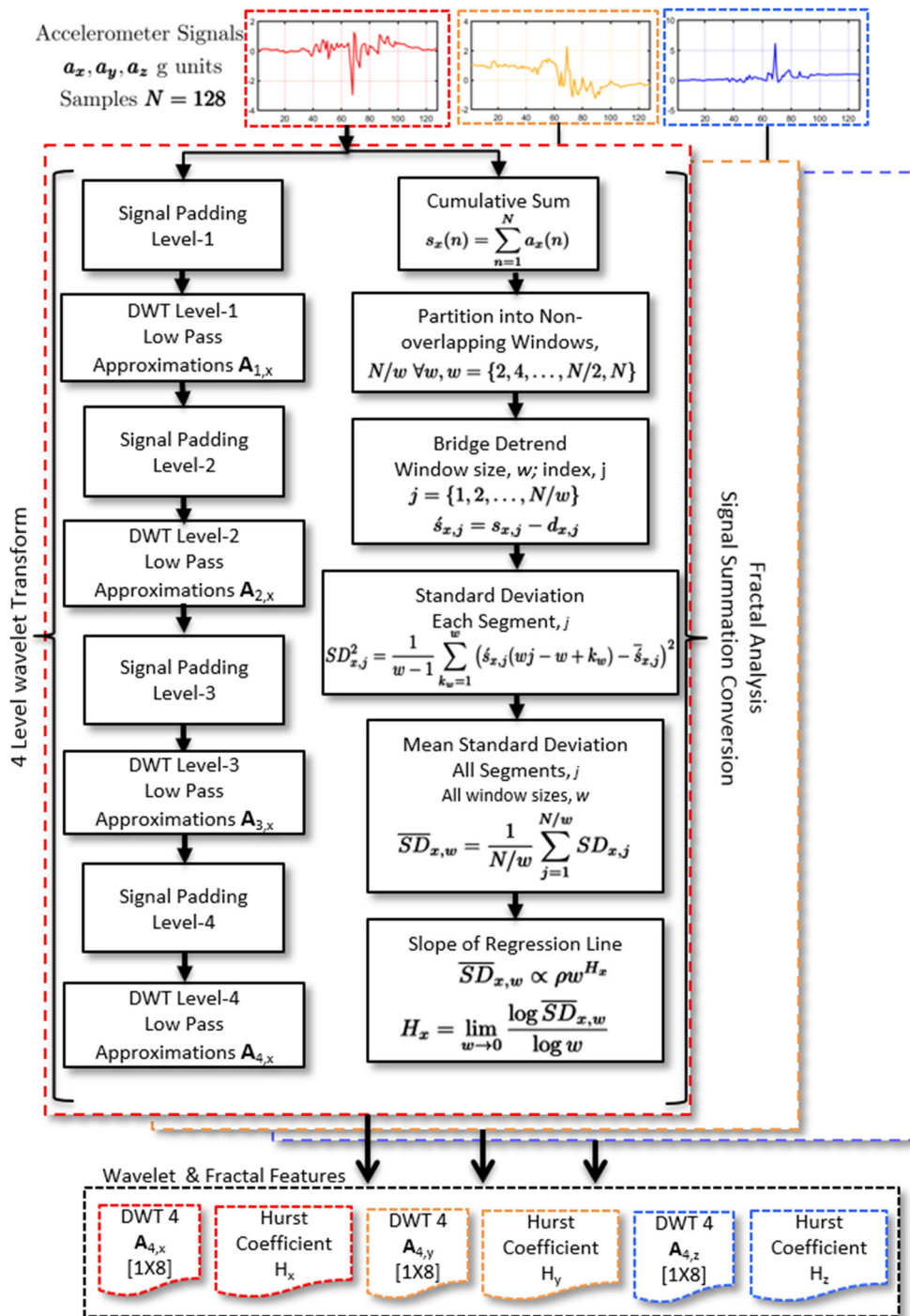


Fig. 5 Proposed fractal features algorithm for the ensemble

Theoretical Analysis of RVFL Ensemble and Heuristic Selection Algorithm

Given an r -dimensional input space X_{in} with input data set \mathbf{x}_{di} and outputs y_{di} with output space Y_{out} representing classes $c = \{c_1, c_2\}$, where c_1 and c_2 are fall and no fall classes, respectively. The set $di = \{1, \dots, N_i\}$ represents the data index and $r = 27$ are the number of wavelet

and fractal features. The DWT and fractal features for all the three axes of motion are used as inputs into the RVFL neural networks. The RVFL models for the ensemble are trained with different activation functions and inputs for heterogeneity and the training is achieved with a closed form solution in Eq. 22. The training set is given as $S_t = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_{N_t}, y_{N_t})\}$. The RVFL classifiers are selected from different folds in the k -fold training

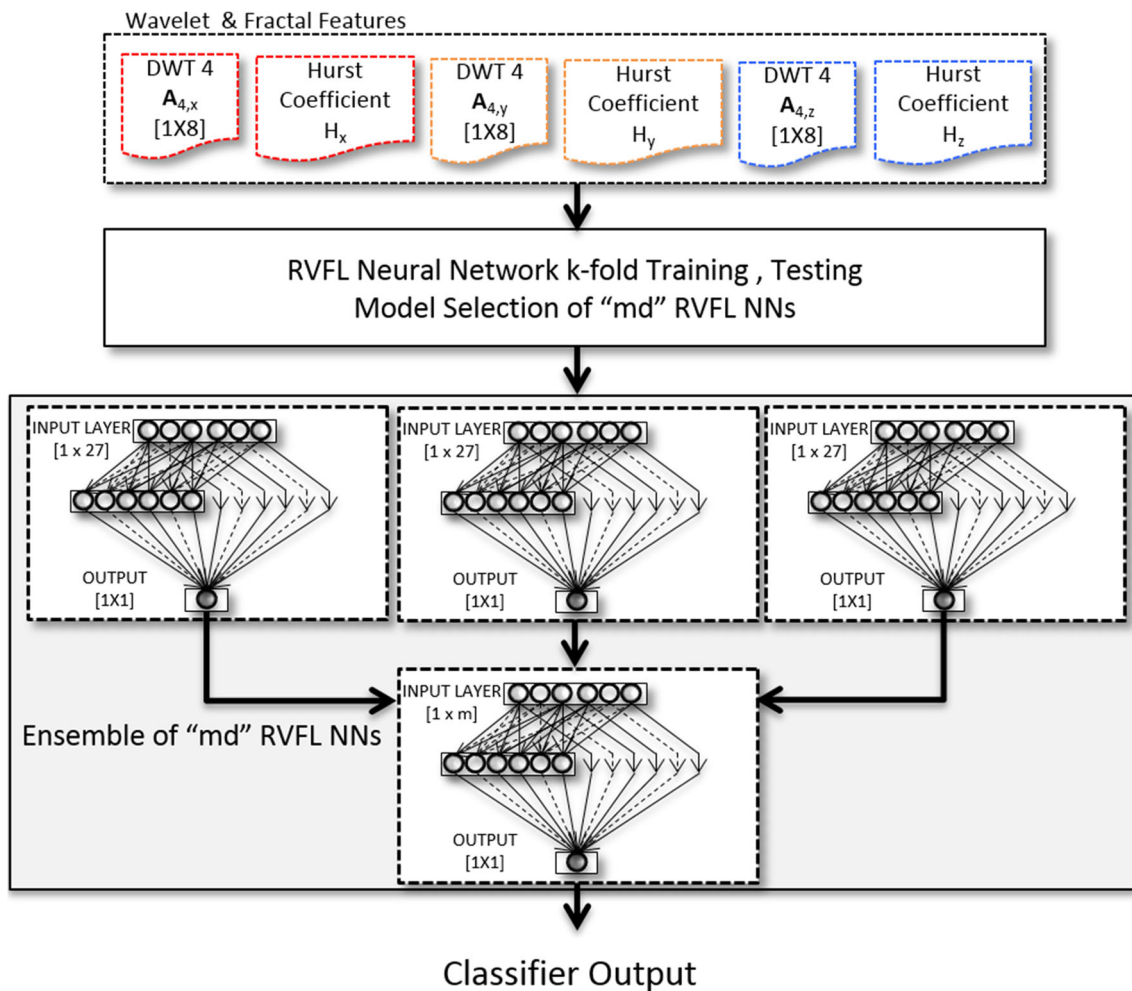


Fig. 6 Proposed RVFL ensemble algorithm with fractal feature inputs

procedure for diversity in training data and low runtime costs of selecting a number of base classifiers from the same training procedure. The training procedure is repeated with three best performing activation functions sine, radbas and tribas for training of heterogeneous base classifiers.

Given L base RVFL classifier models with $M = \{M_1, M_2, \dots, M_L\}$ each model $M_l : X_{in} \rightarrow Y_{out}$ maps input to output space and is trained, such that:

$$\forall l, M_l(\mathbf{x}_{di}) = \arg \max_{c_k \in c} p_{l,c_k}(\mathbf{x}_{di}) \tag{23}$$

where, the function $p_{l,c_k}(\mathbf{x}_{di})$ is the performance score returned by each classifier M_l for input \mathbf{x}_{di} and target class $c_k \in c$. We utilise accuracy measure for each classifier and the performance score can be given as:

$$p_l(\mathbf{x}_{di}) = 1 - \left[f_{act,l}(\boldsymbol{\beta}_{x,l} \cdot \mathbf{x}_{di}^T + \boldsymbol{\beta}_{g,l} \cdot y_g(\mathbf{x}_{di}^T)) - t \right] \tag{24}$$

where t is target output, $f_{act,l}$ is the activation function for model l , y_g is given in Eq. 15, $\boldsymbol{\beta}_{x,l}$ and $\boldsymbol{\beta}_{g,l}$ can be obtained from Eq. 21 and l denotes unique values of the weight vector

for each base classifier, given as:

$$\boldsymbol{\beta}_{x,l} = \begin{bmatrix} \beta_{1,l} \\ \vdots \\ \beta_{r,l} \end{bmatrix}, \tag{25}$$

$$\boldsymbol{\beta}_{g,l} = \begin{bmatrix} \beta_{r+1,l} \\ \vdots \\ \beta_{r+G,l} \end{bmatrix} \tag{26}$$

Equation 23 can now be represented as:

$$\forall l, M_l(\mathbf{x}_{di}) = \arg \max_{di} \sum p_l(\mathbf{x}_{di}) / N_t \tag{27}$$

The RVFL ensemble algorithm seeks a RVFL meta-learner $M_c : X'_c \rightarrow Y_{out}$, where X'_c represents the output space of base classifiers, $\{M_1(x_{di}), M_2(x_{di}), \dots, M_L(x_{di})\}$ and takes the class outputs from base models as input features for the meta-learner. The final base models are selected from the available models based on accuracy and a pairwise diversity indicator, which is used as a heuristic function. Selection of diverse RVFL base

classifiers must be based on an efficient technique in order to benefit from faster learning advantage of base classifiers, while finding diversity amongst the models from individual inputs and their corresponding classification outputs is not computationally feasible for comparing a large number of models. We propose an ensemble selection method based on diversity of models from overall performance measures TP and TN values of base classifiers. The RVFL models for the ensemble are selected based on a diversity indicator computed from the differences in TP and TN values. The insight is based on the observation that two models with the same accuracy can be inherently diverse based on whether the accuracy performance has been achieved through higher TP or higher TN values. The high accuracy of a model may be attributed to its higher TP values (or lower false negative values), while another model with comparable accuracy may have comparatively higher TN values (or lower false positive values), since accuracy depends on the sum of TP and TN values. The two models will complement each other in an ensemble since different TP and TN values also imply that they will differ in their classification or misclassification of instances. The difference in TP and TN values can be a potential indicator of diversity. Therefore, a model with a relatively lower accuracy may be chosen based on a higher TP or TN value.

The models in $M = \{M_1, M_2, \dots, M_L\}$ are sorted based on accuracy metric and a search algorithm based on the diversity indicator as a heuristic function is performed. The resulting set of models obtained, $E = \{M_1, M_2, \dots, M_{md}\}$, consists of md diverse models obtained from heuristic search from a pairwise diversity heuristics. The base model selection algorithm to select md RVFL base classifiers is illustrated in Fig. 7 and is given as follows:

1. Index and sort the base classifier models based on accuracy, $\mathbf{M} = \{M_1, M_2, \dots, M_L\}$.
2. Select the first model M_1 with the highest accuracy in \mathbf{M} and add to the ensemble set, $\mathbf{E} = \{M_{es}\}$. The variable es is used to keep track of the selected number of ensemble learners.
3. Increment count cnt , count checks, if all the models have been tested for comparison with the selected model.
4. Compute the difference in TP and TN values to determine if accuracy values have been achieved through higher TP or TN values. Let M_A be the newly added model to the ensemble set (the highest accuracy model in the start) and M_B be the next model in the sorted model set.

$$\Delta TP = TP_B - TP_A \quad (28)$$

$$\Delta TN = TN_B - TN_A \quad (29)$$

5. Compute the diversity indicator APDI as given in Eq. 30 to check if the next highest accuracy model has either TP or TN value greater than the newly added model to the ensemble set.

$$APDI = \max(\Delta TP, 0) + \max(\Delta TN, 0) \quad (30)$$

6. If APDI is greater than zero, then add the model to the ensemble; otherwise, test the next model for diversity.
7. If no model is found according to the APDI criteria, then add the next highest accuracy model to the ensemble set and repeat the procedure by comparing other models with the next newly added model.

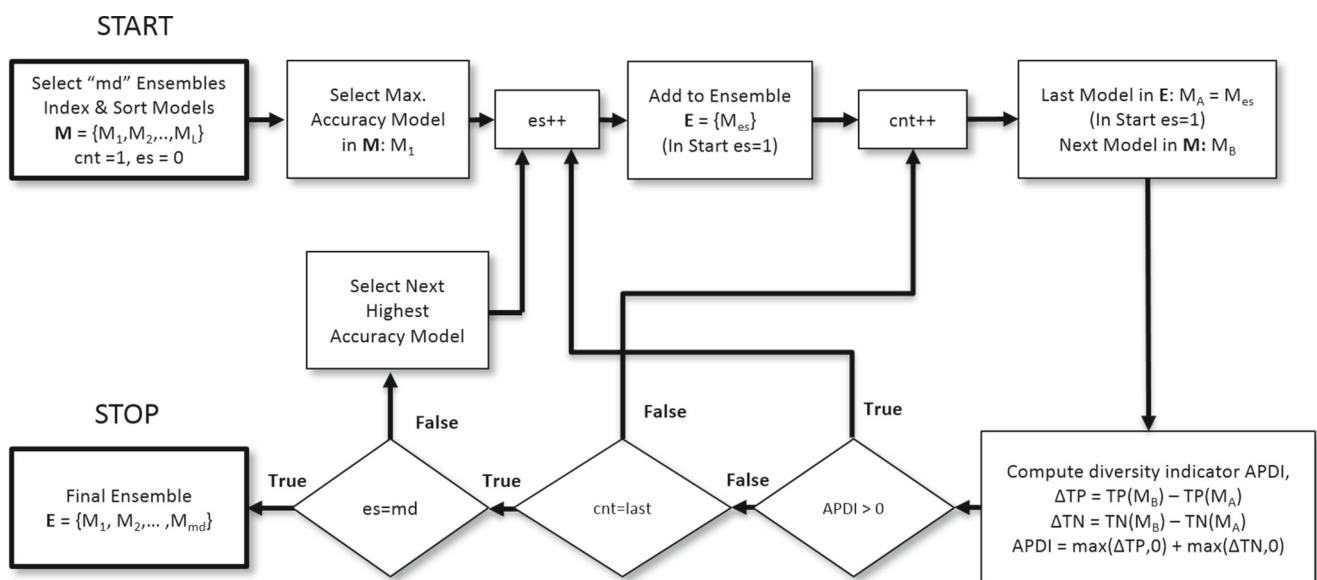


Fig. 7 Ensemble selection algorithm

8. Stop the selection after the required number of ensemble learners is achieved, the final ensemble is given as $E = \{M_1, M_2, \dots, M_{md}\}$.

The ensemble of md selected RVFL neural networks is then combined with another RVFL network with md input neurons equal to the number of RVFL models used in the ensemble. The final RVFL ensemble is a RVFL meta-learner trained with the closed form solution given in Eq. 22 and maps $M_c : X'_c \rightarrow Y_{out}$, where X'_c represents the output space of the selected diverse base classifiers, $E = \{M_1(\mathbf{x}_{di}), M_2(\mathbf{x}_{di}), \dots, M_{md}(\mathbf{x}_{di})\}$ and takes the class outputs from the selected base models as input features for the meta-learner. The RVFL meta-learner is trained, such that:

$$M_c(\mathbf{M}(\mathbf{x}_{di})) = \arg \max_{c_k \in c} p_{c_k}(\mathbf{M}(\mathbf{x}_{di})) \quad (31)$$

where, the function $p_{c_k}(\mathbf{M}(\mathbf{x}_{di}))$ is the performance score returned by the RVFL meta-learner M_c for input $\mathbf{M}(\mathbf{x}_{di})$ and target class $c_k \in c$, where $\mathbf{M}(\mathbf{x}_{di})$ represents the output of base classifiers. We utilise accuracy measure for the RVFL meta-learner classifier and the performance score can be given as:

$$p(\mathbf{M}(\mathbf{x}_{di})) = 1 - \left[f_{act,c}(\boldsymbol{\beta}_{x,c} \cdot \mathbf{M}(\mathbf{x}_{di})^T + \boldsymbol{\beta}_{g,c} \cdot y_g(\mathbf{M}(\mathbf{x}_{di})) - t \right] \quad (32)$$

where t is target output, $f_{act,c}$ is the activation function for meta-learner, y_g is given in Eq. 15, and $\boldsymbol{\beta}_{x,c}$ and $\boldsymbol{\beta}_{g,c}$ are weight vectors for the RVFL meta-learner similar to Eqs. 25 and 26.

Methodology

Dataset

A dataset by Kwolek et al. [29] consisting of accelerometer signals for falls and various ADLs including walking, sitting down, sitting on chair, lying down, lying on bed, picking up objects, standing up and sitting down was used for analysis and experimental verification of the proposed scheme. A total of 40 fall activities were recorded. The dataset is acquired from a motion sensing platform consisting of an Inertial Measurement Unit (IMU) mounted on the pelvis of 5 volunteers. The IMU consists of two sensors, a 16-bit three-axis gyroscope and a 12-bit three-axis accelerometer with a total sampling rate of 256 Hz. The accelerometer was used for analysis and detection of fall activity in this work. The three-axis accelerometer measured the acceleration of the body movements along all the three axes of motion in units of G-force (g) with values varying from -8 to $8g$. All

the three axes of motion are used for classification and are divided into windows of 128 samples each. The 128-sample windows were overlapped during training with 64 samples from the previous window, resulting in a 50% overlap between two windows. Figure 8 shows 128 sample segments for each of the three axes of motion for fall activity. Each of the three axes was processed separately by the algorithm and wavelet and fractal features were computed for each motion axis separately. The features for reach axis were then concatenated and used for training the RVFL ensemble.

Experimental Specifications

This section discusses the experimental specifications including the tools and the system used for processing. The fractal analysis was performed in R language with R-Studio and a number of R packages for fractal analysis were used, including “fractal”, “fracdiff” and “tseries” packages. After fractal analysis in R, the fractal features were stored in data files and exported to Matlab for ensemble classification. The classification and machine learning experiments were performed with the statistics and machine learning toolbox in Matlab version 2019a. The machine learning classification was performed on an Intel system i5-6500 processor with quad cores at 3.2GHz with 6MB cache and a main memory of 8GB. A number of classifiers were used for comparison including, DT, Linear Discriminant Analysis (LDA), KNN, SVM, RF and ANN. The classifiers are explained in “Classifiers” section, while the training and testing strategy is based on a 5 fold partitioning strategy further explained in “*k*-Fold Partitioning” section. The execution runtimes were obtained by reading the clock with Matlab commands “tic” and “toc”. The clock times in cycles before and after the execution of the algorithm were saved and the difference was computed to measure the execution time of the algorithm. A total of 5 measurements were taken for all the 5 combinations of 5-fold partitioning strategy and then an average training runtime was calculated for each algorithm. Furthermore, the system is also compared with current state-of-the-art ensemble techniques utilised for fall detection in “Results and Discussion” section and Table 7.

k-Fold Partitioning

The datasets for training all the classifiers were divided into 5 folds, with 4 folds for training and 1 fold for testing resulting in a percentage of 80–20 % for training and testing, respectively. All the classifiers were trained on the 4 out of 5 folds each time in a round robin fashion and the values for testing accuracy were averaged overall. Similarly, TP/TN, FP/FN and values for precision, sensitivity, specificity and F1-measure were calculated for each fold and averaged

overall. The same strategy was followed for measuring the training time. The training time was measured for training over 4 folds each time and averaged. The specific details of parameters for each classifier are mentioned in the next section.

Classifiers

A number of classifiers with the proposed set of features, namely Daubechies 4, level-4 wavelet coefficients and generalised Hurst exponents, were used for comparison with the proposed RVFL stacking ensemble. The classifiers along with their parameters are given below:

Decision Tree

The DT algorithm was tested on Daubechies 4, level-4 wavelet coefficients and generalised Hurst exponent features of the accelerometer signal. The features are compared against constant values and the tree is split based on less than equal to or greater than value from the constant. Leaf nodes give the final classification of a fall or an ADL decision. The DT in this work uses the CART algorithm to select the best split feature at each node from fractal features and the 4th level wavelet coefficients. Gini's Diversity Index (G.I.) in Eq. 33 is used as the split criterion, with $r - 1$ maximum splits where r is the feature set size or input sample size from fall and activities dataset.

$$G.I. = 1 - \sum_{di=1}^{N_t} pr^2(di) \quad (33)$$

Where pr represents the probability value. The leaves originating from the same parent node are merged and the classification tree is grown by estimating a pruned optimal sequence of subtrees. For the testing phase, the test feature set is routed down the tree according to the values of the features, which are compared against constants at each node and the final classification is obtained on reaching a leaf node associated with a fall or an ADL class.

Linear Discriminant Analysis

LDA finds the maximum separation between classes by maximising variance between classes and minimising variance within the class. Given a set of fractal and wavelet transform features as input $\mathbf{x}_{di} \in R^r$, $di = \{1, \dots, N_t\}$ for $r = 27$ dimensional input space with K classes, labelled as $\{\prod_1, \dots, \prod_K\}$. The k^{th} class has total N_k inputs, with $\mathbf{x}_{di} \in \prod_k$ in the feature space. LDA finds basis vector θ in terms of the between class scatter matrix \mathbf{S}_B and within

class scatter matrix \mathbf{S}_W as:

$$\theta = \underset{\theta}{\operatorname{argmax}} \frac{\theta^T \mathbf{S}_B \theta}{\theta^T \mathbf{S}_W \theta} \quad (34)$$

$$\mathbf{S}_B = \sum_{k=1}^K N_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T \text{ and} \quad (35)$$

$$\mathbf{S}_W = \sum_{k=1}^K \sum_{\mathbf{x}_{di} \in \prod_k} (\mathbf{x}_{di} - \boldsymbol{\mu}_k)(\mathbf{x}_{di} - \boldsymbol{\mu}_k)^T \quad (36)$$

where $\boldsymbol{\mu}$ is the mean vector of all input datasets of size N_t and $\boldsymbol{\mu}_k$ is the mean vector for class k of size N_k , given as:

$$\boldsymbol{\mu} = \frac{1}{N_t} \sum_{di=1}^{N_t} \mathbf{x}_{di}, \quad \boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{\mathbf{x}_{di} \in \prod_k} \mathbf{x}_{di} \quad (37)$$

K-Nearest Neighbour

The KNN classifier is based on the insight that the class of an unknown instance should be similar to the class of its neighbours. The KNNs are chosen based on the Euclidean distance from an unknown instance and a classification decision is based on the majority vote of the neighbours. Given r , where r is the number of wavelet level-4 coefficients and generalised Hurst exponent features, an r -dimensional space can be used to represent the dataset of falls and activities. The Euclidean distance $dist$ between two points, an unknown activity ax and a known activity bx in an r -dimensional feature space can be given by Eq. 38:

$$dist_{ax,bx} = \left[\sum_{D=1}^r (ax_D - bx_D)^2 \right]^{\frac{1}{2}} \quad (38)$$

where $r = 27$ for our feature space. The Euclidean distances between point a and all classified instances are calculated. The KNNs are selected based on the K smallest Euclidean distances and the neighbours are then majority voted to determine a fall or an ADL classification for the unknown activity. In our work, we used several values of K from 1, 3, 5 to 7. The highest classification accuracy was achieved for $K = 1$ nearest neighbours.

Support Vector Machine

SVM [2, 10] finds the best hyperplane with the largest margin that separates the two classes of fall activity and ADLs. Given a set of training input vectors $\mathbf{x}_{di} \in R^r$, $di = \{1, \dots, N_t\}$ for $r = 27$ dimensional fractal and wavelet transform feature space and outputs $y_{di} \in \{1, -1\}$, the hyperplane is given by Eq. 39:

$$y_{di} = \operatorname{sign}(\mathbf{w} \cdot \mathbf{x}_{di}^T + b) \quad (39)$$

where \mathbf{x} and \mathbf{w} represent column vectors of input variables and constants in the hyperplane equation, respectively.

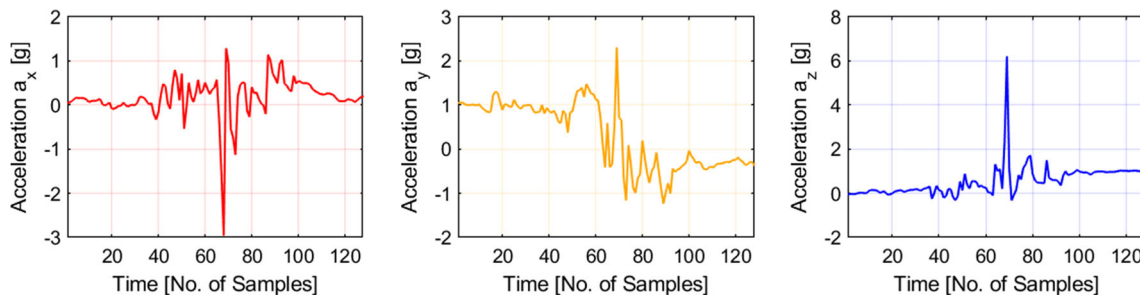


Fig. 8 Tri-axis accelerometer signals

While the training input vector \mathbf{x}_{di} represents the fractal and wavelet transform features and $sign()$ is a signum function with ± 1 output, we use soft-SVM in our Matlab implementation, since it also applies to non-linearly separable data. The objective in soft-SVM is to minimise Eq. 40:

$$\min_{w,b,\xi} \frac{1}{2} \|\mathbf{w}\|^2 + C_b \sum \xi_{di} \quad (40)$$

$$\text{(subject to)} \quad y_{di}(\mathbf{w}^T \mathbf{x}_{di} + b) \geq 1 - \xi_{di} \quad (\forall di)$$

$$\xi_{di} \geq 0 \quad (\forall di)$$

where ξ_{di} is the slack variable and penalises the objective function for data points that cross margin boundary meant for that class, while C_b is the box constraint. We used the Sequential Minimal Optimisation [13] solver in Matlab with a linear kernel function for training the Soft-SVM and a value of 1 for the box constraint.

Random Forest

RF is an ensemble learning technique that generates a number of DTs at training time and outputs the mode of the classes as the final classification output. Given a set of fractal and wavelet transform features as input $\mathbf{x}_{di} \in R^r$, $di = \{1, \dots, N_t\}$ and outputs $y_{di} \in \{0, 1\}$, RF randomly selects inputs with replacement and trains classification tree, also selecting randomly a subset of features at each split. The splitting criterion is either based on the information gain or Gini’s index given in Eq. 33.

Artificial Neural Network

The ANN used is a classic MLP. For a given input vector \mathbf{x}_i , the output of each neuron is computed as:

$$z = \mathbf{w} \cdot \mathbf{x}_{di}^T + b \quad (41)$$

$$f_{sig}(z) = \frac{1}{1 + \exp^{-z}} \quad (42)$$

where f_{sig} represents a sigmoid activation function. The ANN used is a single hidden layer neuron with one input and

one output layer. It was trained and tested with a number of neurons in the hidden layer and various learning algorithms were used including Stochastic Gradient Descent (SGD), Rprop and Levenberg-Marquardt (LM) algorithms.

Results and Discussion

The proposed use of fractal features for classification of falls is a good discriminant along with DWT features and provides high classification accuracies with a number of classifiers as illustrated in Table 6. The proposed RVFL ensemble has a significant speed advantage of the order of 100× and training time of 1.76 ms, which has implications for real time, embedded implementation on low-end processing cores in terms of runtime cost. This would enable real-time detection and immediate notification to medical aid centre for medical response. The proposed stacking ensemble of RVFLs combined with an RVFL meta-learner, along with the proposed ensemble selection algorithm, provides the best results with the proposed features. The RVFL neural networks are first trained with a different number of neurons and activation functions to determine the best parameters for the fall classification problem. Five different activation functions namely, hardlim, sign, sine, tribas and radbas were intially tested. The three activation functions sine, tribas and radbas provide the best results as illustrated in Tables 1 and 2. The number of neurons required for 27 inputs are also modest and good results are achieved with half or less than half the number of total input and output neurons. The table rows with accuracy values highlighted in bold give the best accuracy results. The RVFL networks with the highest accuracies are chosen for the ensemble. However, a network with a higher TP or higher TN is preferred amongst the networks with similar accuracy. For example, in Tables 1 and 2, the three highlighted rows in italic show networks which complement each other either with a higher TP or a higher TN (same is not true for ANN Tables 3 and 4). The ensemble is composed out of three RVFL networks and provides the highest accuracy of 95.71% as illustrated in comparison or results Tables 5 and 6.

Table 1 RVFL neural networks true/false classifications

Neurons	Act. Func.	TP (%)	TN (%)	FP (%)	FN (%)
10	sine	35.71	52.86	4.29	7.14
14		37.14	54.28	2.86	5.71
28		37.14	51.43	5.71	5.71
42		35.71	52.86	4.29	7.14
56		34.29	51.43	5.71	8.57
10	tribas	35.71	55.71	1.43	7.14
14		38.57	52.86	4.29	4.29
28		38.57	51.43	5.71	4.29
42		38.57	51.43	5.71	4.29
56		38.57	52.86	4.29	4.29
10	radbas	37.14	52.86	4.29	5.71
14		37.14	52.86	4.29	5.71
28		37.14	52.86	4.29	5.71
42		37.14	52.86	4.29	5.71
56		35.71	52.86	4.29	7.14

Diverse base networks with similar accuracy but different TP and TN values

The proposed scheme is compared with the most optimised ANN (MLP) with sigmoid activation functions for classification. The ANN is optimised for a number of neurons in the hidden layer and a number of learning algorithms. The three training algorithms stand out for better performance namely, LM, Rprop and SGD. The best training accuracy of 90% is achieved with the LM learning algorithm and 42 neurons followed closely by an ANN

Table 2 RVFL neural networks results

Neurons	Act. Func.	Acc. (%)	Prec. (%)	Sens. (%)	Spec. (%)	F1-Score (%)
10	sine	88.57	89.29	83.33	92.50	86.21
14		91.43	92.86	86.67	95.00	89.66
28		88.57	86.67	86.67	90.00	86.67
42		88.57	89.29	83.33	92.50	86.21
56		85.71	85.71	80.00	90.00	82.76
10	tribas	91.43	96.15	83.34	97.50	89.29
14		91.43	90.00	90.00	92.50	90.00
28		90.00	87.10	90.00	90.00	88.52
42		90.00	87.10	90.00	90.00	88.52
56		91.43	90.00	90.00	92.50	90.00
10	radbas	90.00	89.66	86.67	92.50	88.14
14		90.00	89.66	86.67	92.50	88.14
28		90.00	89.66	86.67	92.50	88.14
42		90.00	89.66	86.67	92.50	88.14
56		88.57	89.29	83.33	92.50	86.21

Table 3 ANN true/false classifications

Neurons	Train	TP (%)	TN (%)	FP (%)	FN (%)
10	LM	37.14	52.86	4.29	5.71
14		31.43	51.43	5.71	11.43
28		34.29	52.86	4.29	8.57
42		37.14	52.86	4.29	5.71
56		35.71	52.86	4.29	7.14
10	Rprop	32.86	52.86	4.29	10.00
14		34.29	52.86	4.29	8.57
28		34.29	51.43	5.71	8.57
42		34.29	52.86	4.29	8.57
56		32.86	51.43	5.71	10.00
10	SGD	34.29	51.43	5.71	8.57
14		32.86	51.43	5.71	10.00
28		35.71	51.43	5.71	7.14
42		34.29	51.43	5.71	8.57
56		34.29	51.43	5.71	8.57

network with 56 hidden neurons, as illustrated in Table 4 with the corresponding TP, TN, FP and FN values in Table 3.

The proposed RVFL ensemble provides the highest accuracy, precision, sensitivity and specificity by virtue of the highest improvement in TP. The accuracy is 5.71% higher than RF and ANN, while it shows an improvement of 7.14% over the SVM and 11.42% over the DT and LDA as illustrated in Table 6.

The proposed scheme gives the lowest training time of 1.76 ms as illustrated in a logarithmic plot in Fig. 9. The

Table 4 ANN parameter tuning for fall classification

Neurons	Train	Acc. (%)	Prec. (%)	Sens. (%)	Spec. (%)	F1-Score (%)
10	LM	90.00	89.66	86.67	92.50	88.14
14		82.86	84.62	73.33	90.00	78.57
28		87.14	88.89	80.00	92.50	84.21
42		90.00	89.66	86.67	92.50	88.14
56		88.57	89.29	83.33	92.50	86.21
10	Rprop	85.71	88.46	76.67	92.50	82.14
14		87.14	88.89	80.00	92.50	84.21
28		85.71	85.71	80.00	90.00	82.76
42		87.14	88.89	80.00	92.50	84.21
56		84.29	85.19	76.67	90.00	80.70
10	SGD	85.71	85.71	80.00	90.00	82.76
14		84.29	85.19	76.67	90.00	80.70
28		87.14	86.21	83.33	90.00	84.75
42		85.71	85.71	80.00	90.00	82.76
56		85.71	85.71	80.00	90.00	82.76

Table 5 Comparison of proposed scheme true/false classifications

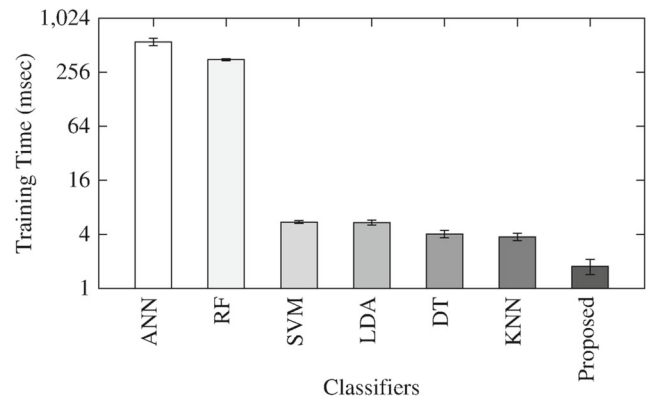
Classifier	TP (%)	TN (%)	FP (%)	FN (%)
DT	34.29	50.00	7.14	8.57
LDA	31.43	52.86	4.29	11.43
KNN	38.57	51.43	5.71	4.29
SVM	35.71	52.86	4.29	7.14
RF	37.14	52.86	4.29	5.71
ANN	37.14	52.86	4.29	5.71
Proposed ensemble	41.43	54.29	2.86	1.43

proposed technique is $2.3\times$ faster than a DT algorithm, which has a training time of 4.06 ms. The improvements in training time are shown in Fig. 10 w.r.t. the ANN. The speedup is considerable with $317.7\times$ that of an ANN and $198.56\times$ that of an RF ensemble method.

The comparison Table in 7 compares the proposed system with state-of-the-art works. The proposed technique gives higher classification performance than the latest works in [36, 58] and [27]. The presented FDS has 8% higher accuracy and 19% higher sensitivity than the camera-based FDS proposed by Wang et al. [58]. Moreover, the proposed technique gives 2.3% and 1.66% higher sensitivity than [36] and [27], respectively. Our work has comparable performance to Chen et al. [6] and 2% lower performance than Chelli et al. [5]. However, the proposed system has the lowest runtime cost of all the recent works illustrated in Table 7. The AE ensemble suffers from higher runtime costs due to the complexity of deep AE networks. The work in [6] has a significant runtime cost at 1810.20 s compared with our RVFL ensemble which takes only 1.76 ms for training and generation phase of RVFL stacking ensemble, while the runtime cost for AE ensemble in Khan et al. [27] is not given, deep AE ensemble is computationally expensive and RVFL networks are computationally fast and efficient learners in shallow networks [43]. The proposed scheme is $2.3\times$ faster than a single DT and tree-based ensemble techniques such as RF, Bagged Trees or EBT require a

Table 6 Comparison results of proposed scheme with different classifiers

Classifier	Acc. (%)	Prec. (%)	Sens. (%)	Spec. (%)	F1-Score (%)
DT	84.29	82.76	80.00	87.50	81.36
LDA	84.29	88.00	73.33	92.50	80.00
KNN	90.00	87.10	90.00	90.00	88.52
SVM	88.57	89.29	83.33	92.50	86.21
RF	90.00	89.66	86.67	92.50	88.14
ANN	90.00	89.66	86.67	92.50	88.14
Proposed ensemble	95.71	93.54	96.66	95.00	95.08

**Fig. 9** Training execution time on log scale

number of DTs for ensemble classification. The tree-based ensembles by Nguyen et al. [36], Wang et al. [58] and Chelli et al. [5] have a lower bound of $\geq 2.3\times 1.76$ ms. The actual runtime costs for the above-mentioned tree classifiers are expected to be much higher than our proposed scheme.

Conclusion

In this paper, we proposed a novel algorithm for classification of falls though the use of fractal features and an ensemble of RVFLs combined with an RVFL neural network. The fractal Hurst exponent is computed from the SSC method and provides an irregularity measure of the signal. The proposed features based on fractal analysis provide a high classification accuracy with DT, LDA, KNN, SVM, RF and ANN and the proposed ensemble. The proposed ensemble utilises a novel and fast selection methodology for base classifiers based on a diversity indicator obtained from overall performance measures of TP and TN values determined during the training procedure. The novel RVFL ensemble classifier proposed in our work gives the highest accuracy of 95.71% compared with other classifiers on the same set of features, an improvement of 5.71% over RF and

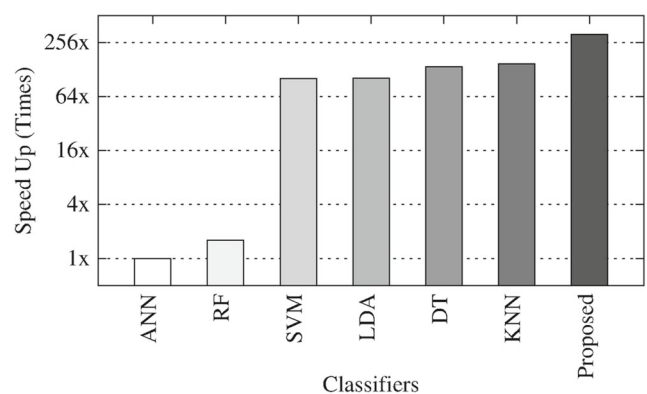
**Fig. 10** Speedup of proposed RVFL ensemble on log scale

Table 7 Comparison with state-of-the-art work

Authors	Khan et al. [27]	Nguyen et al. [36]	Wang et al. [58]	Chelli et al. [5]	Chen et al. [6]	Proposed FDS
Year	2017	2018	2019	2019	2019	2020
Dataset	Datasets [16], [40]	Self-simulated	Dataset [8]	Public datasets	Self-generated	Dataset [29]
Sensor	Tri-axes Acc., Gyro.	Tri-axes Acc., Gyro.	Camera	Tri-axes Acc., Gyro.	Tri-axes Acc.	Tri-axes Acc.
Sensor location	Waist + thigh	Hip	Wall	Thigh + chest	Wrist	Pelvis
Features	x , y and z axes, acceleration and angular velocity	Mean, σ , Energy, entropy, Hjorth mobility, sum vector, correlation, etc.	Image	Mean, maxima, minima, cross Corr., auto Corr., peak PSD, etc.	Mean, Var., crossing rate, Kurtosis, range Freq., skewness	DWT, Hurst exponent
Classifier	AE ensemble	RF	Bagged Tree	EBT	Stacked AE ensemble	RVFL Stack. ensemble
Runtime	-	> 1.76 ms	> 1.76 ms	> 1.76 ms	1810.20 s	1.76 ms
Accuracy	-	-	87.73%	97.70%	-	95.71%
Sensitivity	95%	94.37%	77.11%	-	96.09%	96.66%
Specificity	90%	-	90.76%	-	98.92%	95.00%

ANN, and 7.14% over the SVM. The proposed classifier achieves high gain in runtime. The speedup in training time of the proposed RVFL ensemble is $317.7\times$ compared with an ANN and $198.56\times$ compared with an RF ensemble. The proposed RVFL ensemble is $2.3\times$ faster than a single DT. Furthermore, the proposed scheme has higher or comparable accuracy than most of the latest ensemble methods and provides the lowest runtime cost of 1.76 ms. The proposed ensemble and the ensemble selection algorithm proposed in our work is orthogonal to the application and features used. The speedup advantage of RVFL ensemble can lead to real-time implementation on low-end cores. This can enable on device training and real-time detection and immediate notification for medical response on a fall event.

Compliance with Ethical Standards

Conflict of Interest The authors declare that there are no conflicts of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your

intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bishop CM. Pattern recognition and machine learning. Cham: Springer; 2006.
2. Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. In: Proceedings of 5th annual workshop on computational learning theory. ACM; 1992. p. 144–152.
3. Cannon MJ, Percival DB, Caccia DC, Raymond GM, Bassingthwaite JB. Evaluating scaled windowed variance methods for estimating the Hurst coefficient of time series. *Physica A: Statistical Mechanics and its Applications*. 1997;241(3-4):606–626.
4. Cecotti H. Deep random vector functional link network for handwritten character recognition. In: Proceedings of the international joint conference on neural networks; 2016. p. 3628–3633.
5. Chelli A, Patzold M. A machine learning approach for fall detection and daily living activity recognition. *IEEE Access*. 2019;7:38670–38687.
6. Chen L, Li R, Zhang H, Tian L, Chen N. Intelligent fall detection method based on accelerometer data from a wrist-worn smart watch. *Measurement*. 2019;140:215–226.
7. Chen WH, Ma HP. A fall detection system based on infrared array sensors with tracking capability for the elderly at home. In: 2015 17th international conference on e-health networking, application & services (healthcom). IEEE; 2015. p. 428–434.
8. Chen Z. Fall detection. 2018. URL <https://github.com/kasakun/Fall-Detection>.
9. Consultants PCP market research. Falls: measuring the impact on older people. 2012.

10. Cortes C, Vapnik V. Support-vector networks. *Mach Learn.* 1995;20(3):273–297.
11. Dai P, Gwadry-Sridhar F, Bauer M, Borrie M, Teng X. Healthy cognitive aging: a hybrid random vector functional-link model for the analysis of Alzheimer's disease. In: 31st AAAI conference on artificial intelligence; 2017.
12. Eke A, Herman P, Bassingthwaight J, Raymond G, Percival D, Cannon M, Balla I, Ikrényi C. Physiological time series: distinguishing fractal noises from motions. *Pflügers Archiv.* 2000;439(4):403–415.
13. Fan RE, Chen PH, Lin CJ. Working set selection using second order information for training support vector machines. *J Mach Learn Res.* 2005;6(Dec):1889–1918.
14. Fan X, Zhang H, Leung C, Shen Z. Robust unobtrusive fall detection using infrared array sensors. In: 2017 IEEE international conference on multisensor fusion and integration for intelligent systems. IEEE; 2017. p. 194–199.
15. Feng Q, Gao C, Wang L, Zhao Y, Song T, Li Q. Spatio-temporal fall event detection in complex scenes using attention guided lstm. *Pattern Recogn Lett.* 2020;130:242–249.
16. Frank K, Vera Nadales MJ, Robertson P, Pfeifer T. Bayesian recognition of motion related activities with inertial sensors. In: Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing-Adjunct; 2010. p. 445–446.
17. Gu Y, Zhang Y, Huang M, Ren F. Your WiFi knows you fall: a channel data-driven device-free fall sensing system. In: 2018 5th IEEE international conference on cloud computing and intelligence systems (CCIS). IEEE; 2018. p. 943–947.
18. Hausdorff JM. Gait dynamics, fractals and falls: finding meaning in the stride-to-stride fluctuations of human walking. *Hum Mov Sci.* 2007;26(4):555–589.
19. Hsieh YZ, Jeng YL. Development of home intelligent fall detection IoT system based on feedback optical flow convolutional neural network. *IEEE Access.* 2018;6:6048–6057.
20. Igel'nik B, Pao YH. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Trans Neural Netw.* 1995;6(6):1320–1329.
21. Jahanjoo A, Naderan M, Rashti MJ. Detection and multi-class classification of falling in elderly people by deep belief network algorithms. 2020. *J Ambient Intel Human Comput* 1–21. <https://doi.org/10.1007/s12652-020-01690-z>.
22. Jansi R, Amutha R. Detection of fall for the elderly in an indoor environment using a tri-axial accelerometer and kinect depth data. 2020. *Multidim Syst Sign Process* 1–19. <https://doi.org/10.1007/s11045-020-00705-4>.
23. Jantaraprim P, Phukpattaranont P, Limsakul C, Wongkittisuksa B. Improving the accuracy of a fall detection algorithm using free fall characteristics. In: 2010 International conference on electrical engineering/electronics, computer, telecommunications and information technology. IEEE; 2010. p. 501–504.
24. Katuwal R, Suganthan P, Tanveer M. Random vector functional link neural network based ensemble deep learning. *arXiv preprint arXiv:1907.00350.* 2019.
25. Katuwal R, Suganthan PN, Zhang L. An ensemble of decision trees with random vector functional link networks for multi-class classification. *Appl Soft Comput.* 2018;70:1146–1153.
26. Kerdjiddj O, Ramzan N, Ghanem K, Amira A, Chouireb F. Fall detection and human activity classification using wearable sensors and compressed sensing. *J Ambient Intel Human Comput.* 2020;11(1):349–361.
27. Khan SS, Taati B. Detecting unseen falls from wearable devices using channel-wise ensemble of autoencoders. *Expert Syst Appl.* 2017;87:280–290.
28. Koutsiana E, Hadjileontiadis LJ, Chouvarda I, Khandoker AH. Fetal heart sounds detection using wavelet transform and fractal dimension. *Front Bioeng Biotech.* 2017;5:1–9.
29. Kwolek B, Kepski M. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Comput Methods Prog Biomed.* 2014;117(3):489–501.
30. Lu N, Wu Y, Feng L, Song J. Deep learning for fall detection: 3d-cnn combined with lstm on video kinematic data. *IEEE J Biomed Health Inform.* 2018;3(1):314–23.
31. Maeda K, Takahashi S, Ogawa T, Haseyama M. Convolutional sparse coding-based deep random vector functional link network for distress classification of road structures. *Comput Aided Civ Infrastruct Eng.* 2019;34(8):654–676.
32. Mahoney MW et al. Randomized algorithms for matrices and data. *Found Trends Mach Learn.* 2011;3(2):123–224.
33. Mandelbrot B. How long is the coast of Britain? Statistical self-similarity and fractional dimension. *Science.* 1967;156(3775):636–638.
34. Masud T, Morris RO. Epidemiology of falls. *Age Ageing.* 2001;30:3–7.
35. Melin P, Castillo O. Modelling, simulation and control of non-linear dynamical systems: an intelligent approach using soft computing and fractal theory. Boca Raton: CRC Press; 2001.
36. Nguyen TL, Le TA, Pham C. The Internet-of-Things based fall detection using fusion feature. In: 2018 10th international conference on knowledge and systems engineering (KSE). IEEE; 2018. p. 129–134.
37. Nogas J, Khan SS, Mihailidis A. Deepfall: non-invasive fall detection with deep spatio-temporal convolutional autoencoders. *J Healthcare Inform Res.* 2020;4(1):50–70.
38. Nooruddin S, Islam MM, Sharna FA. An IoT based device-type invariant fall detection system. *Internet Things.* 2020;9:100130.
39. Noury N, Rumeau P, Bourke A, Ó'Laughlin G, Lundy J. A proposal for the classification and evaluation of fall detectors. *Innov Res BioMed Eng.* 2008;29(6):340–349.
40. Ojetola O, Gaura E, Brusey J. Data set for fall events and daily activities from inertial sensors. In: Proceedings of the 6th ACM multimedia systems conference; 2015. p. 243–248.
41. Pang I, Okubo Y, Sturnieks D, Lord SR, Brodie MA. Detection of near falls using wearable devices. *J Geriatr Phys Ther.* 2019;42(1):48–56.
42. Pao Y, Takefuji Y. Functional-link net computing: theory, system architecture, and functionalities. *Computer.* 1992;25(5):76–79.
43. Pao YH, Park GH, Sobajic DJ. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing.* 1994;6(2):163–180.
44. Paul JK, Iype T, Dileep R, Hagiwara Y, Koh JW, Acharya UR. Characterization of fibromyalgia using sleep eeg signals with nonlinear dynamical features. *Comput Biol Med.* 2019;111:103331.
45. Rossignol S, Dubuc R, Gossard JP. Dynamic sensorimotor interactions in locomotion. *Physiol Rev.* 2006;86(1):89–154. DOI: [10.1152/physrev.00028.2005](https://doi.org/10.1152/physrev.00028.2005). <http://www.physiology.org/doi/10.1152/physrev.00028.2005>.
46. Rungnapakan T, Chintakovid T, Wuttidittachotti P. Fall detection using accelerometer, gyroscope & impact force calculation on android smartphones. In: Proceedings of the 4th international conference on human-computer interaction and user experience. ACM; 2018. p. 49–53.
47. Santoyo-Ramón JA, Casilari E, Cano-García JM. Analysis of a smartphone-based architecture with multiple mobility sensors for fall detection with supervised learning. *Sensors.* 2018;18(4):1155.
48. Scardapane S, Wang D. Randomness in neural networks: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery.* 2017;7(2):e1200.
49. Scardapane S, Wang D, Uncini A. Bayesian random vector functional-link networks for robust data modeling. *IEEE Trans Cybern.* 2017;48(7):2049–2059.

50. Sekine M, Tamura T, Akay M, Fujimoto T, Togawa T, Fukui Y. Discrimination of walking patterns using wavelet-based fractal analysis. *IEEE Trans Neural Syst Rehab Eng*. 2002;10(3):188–196.
51. Sukor ASA, Zakaria A, Rahim NA. Activity recognition using accelerometer sensor and machine learning classifiers. In: 2018 IEEE 14th international colloquium on signal processing & its applications. IEEE; 2018. p. 233–238.
52. Terrier P, Dériaz O. Kinematic variability, fractal dynamics and local dynamic stability of treadmill walking. *J Neuroeng Rehab*. 2011;8(1):12.
53. Tian Q, Zhao C, Zhang Y, Qu H. Intrusion signal recognition in OFPS under multi-level wavelet decomposition based on RVFL neural network. *Optik*. 2017;146:38–50.
54. Tian Y, Lee GH, He H, Hsu CY, Katabi D. RF-Based fall monitoring using convolutional neural networks. *Proc ACM Inter Mob Wear Ubiquitous Technol*. 2018;2(3):137.
55. Tian Y, Thompson J, Buck D, Sonola L. Exploring the system-wide costs of falls in older people in Torbay King's Fund. 2013.
56. Wang D. Randomized algorithms for training neural networks. *Inf Sci*. 2016;100(364–365):126–128.
57. Wang D, Li M. Stochastic configuration networks: fundamentals and algorithms. *IEEE Trans Cyber*. 2017;47(10):3466–3479.
58. Wang F, Liu J, Hu G. A novel indoor human fall detection method based on an end-to-end neural network and bagged tree classifier. In: Proceedings of the 2019 2nd international conference on algorithms, computing and artificial intelligence; 2019. p. 384–389.
59. Wang L, Peng M, Zhou Q. Pre-impact fall detection based on multi-source CNN ensemble. *IEEE Sensors J*. 2020;20(10):5442–5451.
60. Wen G, Hou Z, Li H, Li D, Jiang L, Xun E. Ensemble of deep neural networks with probability-based fusion for facial expression recognition. *Cogn Comput*. 2017;9(5):597–610.
61. Xu KK, Li HX, Yang HD. Kernel-based random vector functional-link network for fast learning of spatiotemporal dynamic processes. *IEEE Trans Syst Man Cyber Syst*. 2019;49(5):1016–1026.
62. Xu T, Zhou Y, Zhu J. New advances and challenges of fall detection systems: a survey. *Appl Sci*. 2018;8(3):418.
63. Yang C, Yin XC. Diversity-based random forests with sample weight learning. *Cogn Comput*. 2019;11(5):685–696.
64. Yu X, Qiu H, Xiong S. A novel hybrid deep neural network to predict pre-impact fall for older people based on wearable inertial sensors. 2020. *Front Bioeng Biotechnol*. 8:63.
65. Zhang PB, Yang ZX. A new learning paradigm for random vector functional-link network: Rvfl+. *Neural Netw*. 2020;122:94–105.
66. Zhang YD, Chen XQ, Zhan TM, Jiao ZQ, Sun Y, Chen ZM, Yao Y, Fang LT, Lv YD, Wang SH. Fractal dimension estimation for developing pathological brain detection system based on Minkowski-Bouligand method. *IEEE Access*. 2016;4:5937–5947.
67. Zhou P, Li W, Wang H, Li M, Chai T. Robust online sequential rvflns for data modeling of dynamic time-varying systems with application of an ironmaking blast furnace. *IEEE Trans Cybern* 1–13. 2019. <https://doi.org/10.1109/TCYB.2019.2920483>. (Early Access).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.