Multimedia Tools and Applications
https://doi.org/10.1007/s11042-019-08321-6

# Fuzzy ELM for classification based on feature space

Yonghe Chu[1] · Hongfei Lin[1] · Liang Yang[1] · Dongyu Zhang[1] · Shaowu Zhang[1] · Yufeng Diao[1] · Deqin Yan[2]

Received: 5 December 2018 / Revised: 9 July 2019 / Accepted: 1 October 2019
Published online: 25 July 2020

**Abstract**
As a competitive machine learning algorithm, extreme learning machine (ELM), with its simple theory and easy implementation, has been widely used in the field of pattern accuracy. Recently, researchers have proposed related research algorithms to accommodate noise and outlier data. With a proper fuzzy membership function, a fuzzy ELM can effectively reduce the effects of outliers when solving the classification problem. However, how to apply ELM for learning and accuracy in the presence of noise is still an important research topic**.** A novel fuzzy ELM (ANFELM) technique is proposed in this paper. In the algorithm, the membership degree of the sample is calculated in a feature mapping space instead of the data input space. The algorithm provides good performance in reducing the effects of outliers and significantly improves classification accuracy and generalization. Experiments on UCI datasets and textual datasets show that the proposed algorithm significantly improves the classification capability of ELM and is superior to other algorithms.

**Keywords** Extreme learning machine · Classification · Membership degree · Feature mapping space

## 1 Introduction

Extreme learning machine [8–11, 34](ELM) is proposed by Huang et al. as an extension of traditional single-hidden layer feedforward networks (SLFNs). ELM randomly generates input weights and the offset value of hidden layer nodes. Only the output weights of all the parameters are analysed and determined, and process for solving the traditional neural network is based on a linear model. A traditional neural network algorithm, such as the BP [25] neural network, adjusts the input weight and offset value of the hidden layer nodes by a gradient-

✉ Hongfei Lin
hflin@dlut.edu.cn

[1] Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China

[2] School of Computer and Information Technology, Liaoning Normal University, Dalian 116081, China

based method in an iterative manner. However, gradient descent-based methods have the disadvantages of high time complexity and easily trapping the search in a local optimal solution. Compared with traditional neural network algorithms, ELMs randomly generate the input weights and offset value of the hidden layer nodes. Therefore, ELMs can spend less time getting the optimal solution and require less human intervention in the training process than traditional neural network algorithms. It has been shown that even without updating the parameters of the hidden layer, an SLFN with randomly generated hidden neurons and tuneable output weights maintains its universal approximation capability [10, 37]. Compared to gradient-based algorithms, ELMs are much more efficient and usually lead to better generalization performance. In the literature [2, 7, 9], ELMs have better generalization ability than Support Vector Machine (SVM) [29] and its improved algorithms.

In recent years, ELMs have made great progress in both theory and application. Huang et al. [10] theoretically studied the universal approximation ability of ELM. Gao et al. [16, 17, 19] used statistical learning theory to conduct in-depth research on the generalization ability of ELM. The generalization error bound of ELMs has been investigated from the perspective of the Vapnik–Chervonenkis (VC) dimension theory and the initial localized generalization error model (LGEM) [30]. Empirical studies have shown that the generalization ability of ELM is comparable to or even better than that of SVMs and their variants [2, 12]. In terms of applications, Raghuwanshi et al. [23, 24, 33] studied the defects of ELM in using imbalanced data for ELM algorithm improvements. Liu et al. [20] and Gu et al. [5] used manifold learning technology to apply ELM to semi-supervised learning. Li [28] et al. used the joint training method to apply ELM to semi-supervised learning. Cao et al. [3] and Li et al. [15, 21, 27] applied ELM to process remote sensing images. Liang et al. [22, 26, 41] proposed various improvements for the problems of ELM in online sequential data applications. Recently, researchers have combined ELM and dimensionality reduction techniques. Lavneet et al. [14] and Castaño et al. [4] applied principal component analysis (PCA) dimensionality reduction technology to ELM. Wang et al. [31] combined the local tangent space alignment (LTSA) dimensionality reduction algorithm with ELM.

The above improvements in theory and application enhance the generalization capability of ELM and greatly expand the application range of ELM. However, in many applications, some training points are corrupted by noise. Moreover, some points in the training data are accidentally misplaced. These points are all outliers; thus, they do not all belong to one class. It is important to fully assign some of the points to one class so that ELM can separate these points more correctly. Some data points corrupted by noises are less meaningful than other data points, and the machine should discard them. ELM lacks this kind of ability.

In this paper, we use membership degree to build an ELM model called ANFELM. The degree of membership characterizes the importance of the data samples to the ELM model. Inspired by the literature [13], we assign an additional weight according to the influences of different data samples. The greater the degree of membership is, the greater the impact of data points on the ELM model. Regarding noise and outlier data, a large membership degree is assigned to important data, and a small membership degree is assigned to the noise.

The advantages of ANFELM can be summarized as follows:

1) ANFELM is simple in theory and convenient in implementation, which retains the advantages of the original ELM.

2) The ANFELM algorithm takes into account the influence of the feature mapping space on the data when calculating the membership degree of the sample so that the membership degree of the sample is calculated in the feature mapping space instead of the data input space.

The rest of the paper is organized as follows: related work is introduced in the second part, and ELM is introduced in the third part. The details of the new fuzzy membership function used for ANFELM are discussed in Sect. 3. The experimental results and analysis will be given in the fifth part. The final conclusion is given in the sixth part.

## 2 Related works

Recently, researchers have studied the problems of the ELM algorithm in dealing with noise and outliers. In the literature [18], a robust activation function extreme learning machine (RAFELM) was proposed. RAFELM improves the Gaussian activation function, enhances the ELM ability to resist interference from noise and outliers, and improves the generalization capability of ELM. Zhang et al. [36] used the $\ell_1-$ normal loss function to enhance the robustness of ELM to noise and outliers. The work In the literature [35, 38] used the concept of membership degree to assign a corresponding membership degree according to the different influences of different data samples on the ELM model. In the literature [35], a fuzzy extreme learning machine (FELM) was proposed for classification problems. FELM proposed the concept of membership degree did not specifically suggest how to calculate it. In the literature [38], a new fuzzy extreme learning machine (NFELM) algorithm was proposed for regression problems. In order to solve the problem of noise points in regression problem, NFELM uses the idea of membership degree to assign a membership degree to different samples. Assign a smaller membership to the noise point, thereby reducing the impact of noise points on the ELM algorithm. Xia et al. [32] applied the method of nuclear clustering to ELM for handling unbalanced data and the influence of noise and outliers on ELM, proposing the Possibilistic Fuzzy Extreme Learning Machine (PFELM-CIL) model.

Although the abovementioned studies have used different approaches to make significant contributions to dealing with the problems of noise outliers and improving the robustness of ELM, there is still much work to do. The work in [18] improved activation functions from the perspective of the Gaussian activation function, which enhances the ELM’s capability to avoid interference from noise and outliers. However, RAFELM does not eliminate the influence of noise and outliers on the ELM algorithm. Both [32, 38] used membership degree to improve the ELM algorithm. Reference [38] used the average distance between the sample point $x_i$ and the neighbourhood point as the membership degree of $x_i$ in the input space. Reference [32] uses the concept of clustering to define sample membership. However, references [32, 38] do not take into account the impact of the ELM feature mapping space on the data. The data samples are mapped to the hidden layer node space by the activation function or kernel function, and then the output weights are solved when ELM addresses classification and regression problems. We know that both the activation function and the kernel function are nonlinear mappings such as the Sigmoid activation function and the Gaussian kernel function. In references [32, 38], fuzzy memberships were calculated in the input space but not in the feature space; thus, the contribution of each point to the construction of the hyperplane in the feature space cannot be represented properly. One of the most important aspects of FELM is choosing appropriate fuzzy memberships for a given problem. In this paper, we propose a new fuzzy membership function for ELMs. We calculate fuzzy membership in the feature space and represent it with an activation function. We will verify the robustness of the proposed algorithm for noise and outliers in the experiment.

## 3 ELM

In this section, we briefly review the extreme learning machine algorithm as a preliminary description of our work. The extreme learning machine proposed by Huang et al. [8] is an efficient and practical learning mechanism for single-layer feedforward neural networks. The network structure of extreme learning machine is shown in Fig. 1.

For $N$ different samples $(x_j, t_j)$, input can be expressed as $X = (x_1, x_2, \cdots, x_N)^T \in R^{D \times N}$, where $t_j = (t_{j1}, t_{j2}, \cdots, t_{j\mathrm{m}})^T \in R^m$, and the ELM model with the$L$ hidden layer node activation function $g(x)$ is as follows:

$$\sum_{i=1}^{L} \beta_i g\left(a_i \cdot x_j + b_i\right) = t_j \tag{1}$$

where $j = 1, 2, \cdots, N$, $a_i = (a_{i1}, a_{i2}, \cdots, a_{iD})$ are the input weight vectors connecting an $i$th hidden layer node and an input node, and $\beta_i = (\beta_{i1}, \beta_{i2}, \cdots, \beta_{im})$ is the output weight vector connecting the $i$th hidden layer node and the output node. $b_i$ is the offset value of the $i$th hidden layer node. $a_i \cdot x_j$ represents the inner product of $a_i$ and $x_j$. $t_j = (t_{j1}, t_{j2}, \cdots, t_{j\mathrm{m}})^T \in R^m$ is the expected output vector corresponding to the sample $x_j$. To integrate all data samples, (1) can be rewritten as follows:

$$H\beta = T \tag{2}$$

where $H$ is the network hidden layer node output matrix, $\beta$ is the output weight matrix, and $T$ is the expected output matrix:

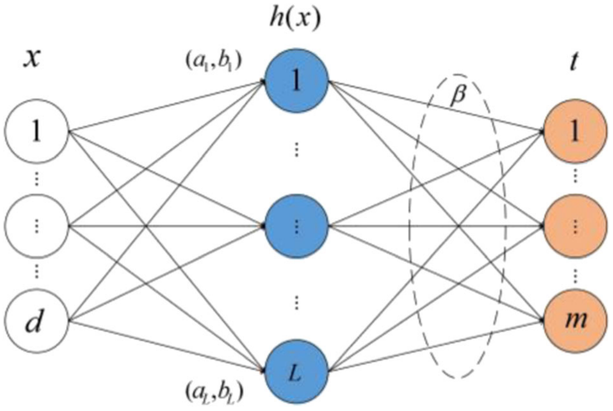


**Fig. 1** The network structure of ELM

$$H = \begin{pmatrix} g(a_1 \cdot x_1 + b_1) & \cdots & g(a_L \cdot x_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(a_1 \cdot x_N + b_1) & \cdots & g(a_L \cdot x_N + b_L) \end{pmatrix}_{N \times L} \tag{3}$$

$$\beta = \begin{pmatrix} {\beta_1}^T \\ \vdots \\ {\beta_L}^T \end{pmatrix}_{L \times m}, T = \begin{pmatrix} {t_1}^T \\ \vdots \\ {t_N}^T \end{pmatrix} = \begin{pmatrix} t_{11} \cdots, t_{1m} \\ \vdots \\ t_{N1} \cdots t_{Nm} \end{pmatrix} \tag{4}$$

When the number of hidden layer nodes is the same as the number of training samples ($L = N$), we can directly obtain the optimal output weight matrix $\beta$ by the inverse of matrix $H$ by (2). However, in most cases, the number of hidden layer nodes is much smaller than the number of training samples ($L < N$). At this time, the matrix$H$is a singular matrix. We solve Eq. (2) by the least squares solution

$$\hat{\beta} = \arg\min_{\beta} \|H\beta - T\|^2 = H^+ T \tag{5}$$

where $H^+ = (H^T H)^{-1} H^T$ is the generalized inverse matrix of the matrix$H$.

To improve the stability and generalization capability of traditional ELM, Huang proposed equality optimization constraint-based ELM [6]. The optimization of the ELM with an equality optimization constraint not only minimizes the training error $\varepsilon$ but also minimizes the output weight $\beta$, so the ELM target with the equality optimization constraint can be written as

$$\min_{W} \frac{1}{2}\|\beta\|^2 + \frac{1}{2} C \sum_{j=1}^{N} \|\varepsilon_j\|^2 \qquad s.t.\ \ h(x_j)\beta = {t_j}^T - {\varepsilon_j}^T \qquad j = 1, 2, \cdots, N \tag{6}$$

in Eq. (6), $\varepsilon_i = (\varepsilon_{i1}, \cdots, \varepsilon_{1m})^T$ is a training error vector corresponding to the sample $x_{j,}$ and $C$ is a penalty parameter.

The number of training samples is larger than the number of hidden layer nodes or the number of training samples is smaller than the number of hidden layer nodes during ELM calculation. The two cases correspond to different output weights, $L = \frac{1}{2}\|\beta\|^2 + \frac{1}{2} C \sum_{j=1}^{N} w_j \|\varepsilon_j\|^2 - \sum_{j=1}^{N} \sum_{i=1}^{L} \alpha_{ji}\left(f(x_j)\beta_i - t_{ji} + \varepsilon_{ji}\right)$. The optimization problem is constrained according to the KKT optimization theory (6) according to the following formula to improve the stability of the ELM:

$$L = \frac{1}{2}\|\beta\|^2 + \frac{1}{2} C \sum_{j=1}^{N} \|\varepsilon_j\|^2 - \sum_{j=1}^{N} \sum_{i=1}^{L} \alpha_{ji}\left(h(x_j)\beta_i - t_{ji} + \varepsilon_{ji}\right) \tag{7}$$

When the number of training samples is less than the number of hidden layer nodes ($L > N$), the solution to (7) is available:

$$\beta = H^T \left( {}^{I}/_{C} + HH^T \right)^{-1} T \tag{8}$$

The output function of ELM obtained by (8) is

$$f(x) = h(x)\beta = h(x)H^T\left(\frac{I}{C} + HH^T\right)^{-1}T \tag{9}$$

When the number of training samples is greater than the number of hidden layer nodes ($L \le N$), the solution to (9) is available:

$$\beta = \left(\frac{I}{C} + H^TH\right)^{-1}H^TT \tag{10}$$

Therefore, the ELM algorithm solving process can be summarized as follows:

1) Initialize the training sample set;
2) Randomly specify the network input weight $a_i$ and the offset value $b_i$,$i = 1, 2, \cdots, L$;
3) Calculate the hidden layer node output matrix $H$ by the activation function;
4) Calculate the output weight $\beta$ according to Eq. (8) or (10) .

## 4 A new fuzzy membership function for ELM

In this section, we propose a new fuzzy membership function. Compared with the method proposed in [38], the proposed method can achieve better results in eliminating the influence of noise and outliers on the ELM algorithm. At the same time, it can significantly improve the accuracy and generalization capability of the ELM algorithm.

### 4.1 Definition of membership

For a given dataset $X = (X_1, X_2, \cdots, X_m, \cdots, X_C)$, $X_m$ is a sample matrix composed of all samples of the $m$th class, and $C$ is the total number of classes in the data samples. $x_i^m$ is the $i$th sample of the $m$th class. The number of data samples of the$m$th class is$n_m$. The total number of data samples is $N$. The dataset $X = (X_1, X_2, \cdots, X_m, \cdots, X_C)$ is mapped from the input space to the hidden layer by the activation function and can be written as $\varphi(X) = (\varphi(X_1), \varphi(X_2), \cdots\varphi(X_m), \cdots, \varphi(X_C))$. We use clustering and $k$ neighbours to define membership. The class centre for all samples of the$m$th class, the radius of all the $m$th class and the distance of samples to the class centre are defined as follows:

$$\varphi^m = \frac{1}{n_m}\sum_{\varphi(x_i)\in\varphi(X_m)} \varphi(x_i) \tag{11}$$

$$R = \max\|\varphi^m - \varphi(X_m)\| \tag{12}$$

$$d_i = \|\varphi(x_i) - \varphi^m\| \tag{13}$$

$$\overline{d_i} = \frac{\sum_{j=1}^{k} \|\phi(x_i) - \phi(x_j)\|}{k} \tag{14}$$

In Eq. (11), $\phi^m$ is the mean of all samples of the $m$th class, $\overline{d_i}$ is the average distance between sample point $x_i$ and neighbour points, and $k$ is the number of neighbours of sample point $x_i$. The above formulas (11) and (12) are applicable to the case where the activation function is known. When the activation function is unknown, we can use the kernel function to find $R$, $d_i$, $\overline{d_i}$:

$$\begin{aligned} R^2 &= \max\|\phi(X_m)-\varphi^m\|^2 \\ &= \max\left\{\varphi^2(X_m)-2\varphi(X_m)\cdot\varphi^m+(\varphi^m)^2\right\} \\ &= \max\left\{(\varphi^m)^2-\frac{2}{n_m}\sum_{i=1}^{n_m}\varphi(x_i)\cdot\varphi(X_m)+\frac{1}{(n_m)^2}\sum_{i=1}^{n_m}\sum_{j=1}^{n_m}\varphi(x_i)\cdot\varphi(x_j)\right\} \\ &= \max\left\{K(X_m,X_m)-\frac{2}{n_m}\sum_{i=1}^{n_m}K(x_i,X_m)+\frac{1}{(n_m)^2}\sum_{i=1}^{n_m}\sum_{j=1}^{n_m}K\left(x_i,x_j\right)\right\} \end{aligned} \tag{15}$$

In Eq. (15), $X_m$ is a sample matrix composed of all samples of the $m$th class, $x_i$ is the $i$th sample of the $m$th class, $n_m$ is the number of the $m$th class data samples, and $K(x_i, x_j)$ is the kernel function.

$$\begin{aligned} d_i^2 &= \|\varphi(x_i)-\varphi^m\|^2 \\ &= \varphi(x_i)^2-2\varphi(x_i)\cdot\varphi^m+(\varphi^m)^2 \\ &= K(x_i,x_i)-\frac{2}{n_m}\sum_{j=1}^{n_m}K\left(x_i,x_j\right)+\frac{1}{(n_m)^2}\sum_{j=1}^{n_m}\sum_{k=1}^{n_m}K\left(x_j,x_k\right) \end{aligned} \tag{16}$$

$$\overline{d_i}^2 = \frac{\sum_{j=1}^{k}\left\|\phi(x_i)-\phi\left(x_j\right)\right\|^2}{k^2} = \frac{\sum_{j=1}^{k}\left\{K(x_i,x_i)+K\left(x_j,x_j\right)-2K\left(x_i,x_j\right)\right\}}{k^2} \tag{17}$$

1) When the activation function is known, the membership degree W is defined by using Eqs. (12), (13), and (14):

$$s_i = 1-\frac{d_i}{R+\sigma} \tag{18}$$

$$\eta_i = 1-\frac{\overline{d_i}}{\overline{\overline{d}_{\max}}+\delta} \tag{19}$$

$$w_i = s_i^{\eta_i} \tag{20}$$

In Eqs. (18) and (19), in order to avoid $s_i = 0$ and $\eta_i = 0$, $\sigma > 0$, $\delta > 0$ and $\overline{d}_{\max} = \max\left(\overline{d_1}, \ldots, \overline{d_N}\right)$, respectively.

2) When the activation function is unknown, we use the kernel function method to define the membership degree $w_i$ by using (14) and (15):

$$s_i = 1-\sqrt{d_i^2 \big/ _{R^2+\sigma}} \tag{21}$$

$$\eta_i = 1-\sqrt{\frac{\overline{d_i}^2}{\overline{d_{\max}}^2 + \delta}} \tag{22}$$

$$w_i = s_i^{\eta_i} \tag{23}$$

In Eqs. (21) and (22), to avoid $s_i = 0$ and $\eta_i = 0$, $\sigma > 0$, $\delta > 0$ and $\overline{d_{\max}}^2 = \max\left(\overline{d_1}^2, \ldots, \overline{d_N}^2\right)$, respectively.

The data samples are mapped to the hidden layer node space by the activation function or kernel function, and then the output weights are solved when dealing with classification and regression problems.

References [32, 38] calculate the membership degree in the input space of the data sample. However, the geometry and properties of the data are different in the ELM feature mapping space, which makes the calculation of membership in the data input space incorrectly reflect the relationship between the data sample and the class Centre and leads to an inaccuracy in the output weight. This paper proposes a new method of determining the membership degree to effectively solve the above problems. The proposed method is applicable not only to the ELM algorithm but also to the nuclear ELM algorithm.

### 4.2 The proposed ANFELM model

We introduce the membership degree $w_i$, which corresponds to each sample obtained in Section 4.1 in the ELM model. By modifying the ordinary ELM formulation (6), we give the formulation of ANFELM as

$$\min_W \frac{1}{2}\|\beta\|^2 + \frac{1}{2}C\sum_{j=1}^{N} w_j\|\varepsilon_j\|^2 \qquad s.t.\ \ h(x_j)\beta = t_j^T - \varepsilon_j^T \qquad j = 1, 2, \cdots, N \tag{24}$$

where $\beta$ is the output weight matrix, $h(x_j) = (g(a_1 \cdot x_j + b_1), \ldots, g(a_L \cdot x_j + b_L))$, $L \times L$ is the training error; and$C$ is the penalty parameter.

The Lagrangian function corresponding to (24) is

$$L = \frac{1}{2}\|\beta\|^2 + \frac{1}{2}C\sum_{j=1}^{N} w_j\|\varepsilon_j\|^2 - \sum_{j=1}^{N}\sum_{i=1}^{L} \alpha_{ji}\left(h(x_j)\beta_i - t_{ji} + \varepsilon_{ji}\right) \tag{25}$$

where $\beta_i$ is the weight $\beta = [\beta_1, \beta_2, \ldots, \beta_m]$ of the $i$th output node, according to the KKT conditions

$$\frac{\partial L}{\partial \beta_i} = 0 \rightarrow \beta_i = \sum_{j=1}^{N} \alpha_{ji} h(x_j)^T \rightarrow \beta = H^T\alpha \tag{26}$$

$$\frac{\partial L}{\partial \varepsilon_j} = 0 \rightarrow \alpha_j = C w_j \varepsilon_j, j = 1, \ldots, N \tag{27}$$

$$\frac{\partial L}{\partial \alpha_j} = 0 \rightarrow h(x_j)\beta - t_j^T + \varepsilon_j^T = 0, j = 1, \ldots N \tag{28}$$

where $\alpha_j = [\alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jm}]^T$ and $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]$.

When the number of training samples is less than the number of hidden layer nodes ($L > N$), we can substitute (20) and (27) into (28) to obtain

$$\left(H^T H + \frac{W}{C}\right)\alpha = T \tag{29}$$

In Eq. (29), $W = diag\left(\frac{1}{w_1}, \frac{1}{w_2}, \cdots, \frac{1}{w_N}\right)$.

From (26) and (29), we can obtain

$$\beta = H^T \left(H H^T + \frac{W}{C}\right)^{-1} T \tag{30}$$

The output function of ANFELM by (30) is

$$f(x) = h(x)\beta = h(x) H^T \left(H H^T + \frac{W}{C}\right)^{-1} T \tag{31}$$

When the number of training samples is greater than the number of hidden layer nodes ($L \leq N$), (26) and (28) can be used to obtain

$$\alpha = \left(H^T\right)^+ \beta \tag{32}$$

$$\varepsilon = \frac{W}{C}\left(H^T\right)^+ \beta \tag{33}$$

Substituting (33) into (28) can result in the following form:

$$\begin{gathered} H\beta - T + \frac{W}{C}\left(H^T\right)^+ \beta = 0 \\ \left\{H - \frac{W}{C}\left(H^T\right)^+\right\}\beta = T \\ W\left(H^T\right)^+ \left\{H^T W^{-1} H + \frac{I}{C}\right\}\beta = T \\ \beta = \left\{H^T W^{-1} H + \frac{I}{C}\right\}^{-1} H^T W^{-1} T \end{gathered} \tag{34}$$

In Eq. (33), $W = diag\left(\frac{1}{w_1}, \frac{1}{w_2}, \cdots, \frac{1}{w_N}\right)$ and $(H^T)^+$ is the generalized inverse matrix of the matrix$H^T$.

The output function of ANFELM by (34) is

$$f(x)=h(x)\beta=h(x)\left\{H^{T}W^{-1}H+\frac{I}{C}\right\}^{-1}H^{T}W^{-1}T \tag{35}$$

given the training sample $X=(x_1,x_2,\cdots,x_N)$ and the expected output matrix $T=(t_1,t_2,\cdots,t_N)\in R^{m\times N}$ of the training samples, $g(x)$ is the activation function. The number of hidden layer nodes is $L$. The ANFELM algorithm can be summarized as follows:

```
ANFELM Algorithm.
Input: $\{(x_j,t_j)\,|\,x_j\in R^d, t_j\in R^m, j=1,2,\ldots N\}$ is the initial training sample set; $g(x)$ is activation function;
The number of hidden layer nodes is $L$;
Output: Output weight matrix $\beta$;
Step 1: Randomly specify the network input weight $a_i$ and offset value $b_i$,  $i=1,2,\cdots,L$;
Step 2: Calculate the hidden layer node output matrix  $F$ by the activation function;
Step 3: Calculate the membership $w_i$ of the sample by using equation (20) or (23);
Step 4:
    If $L<N$
    Compute the output weights $\beta$ using (30)
    Else $L>N$
    Compute the output weights $\beta$ using (34)
return The mapping function $f(x)=h(x)\beta$.
```

## 5 Experimental results and analysis

To prove the effectiveness of the proposed algorithm, we experimented with ANFELM, ELM, NFELM [38], and RAFELM [18] on artificial datasets, UCI datasets and textual datasets. The activation function of ANFELM, ELM, NFELM [38], is sigmoid. RAFELM uses the improved activation function in [18]. For fairness of the experiment, ELM and its improved algorithms use the same penalty parameters (we try to find the optimal penalty parameters to make ELM and its improved algorithms obtain better results). We use MATLAB 2015b to conduct experiments and validate the performance of ANFELM. All experiments were conducted on a computer with an Intel(R) Core(TM) 3.40 GHZ CPU and 8 GB of RAM.

**Table 1** Performance comparison of different ELM algorithms on an artificial dataset

| | ELM | RAFELM | CELM | NFELM | ANFELM |
|---|---|---|---|---|---|
| Accuracy rate(%) | 79.25 | 87.50 | 87.00 | 85.75 | **89.75** |

## 5.1 Experiments with artificial data

First, the artificial data are tested. The training samples and the test samples are randomly generated with two types of two-dimensional noise-containing (noise ratio is 0.2) sample points. The number of training sets is 120, which includes 63 positive examples and 57 negative examples. The number of test sets is 80, which includes 36 positive examples and 44 negative examples.

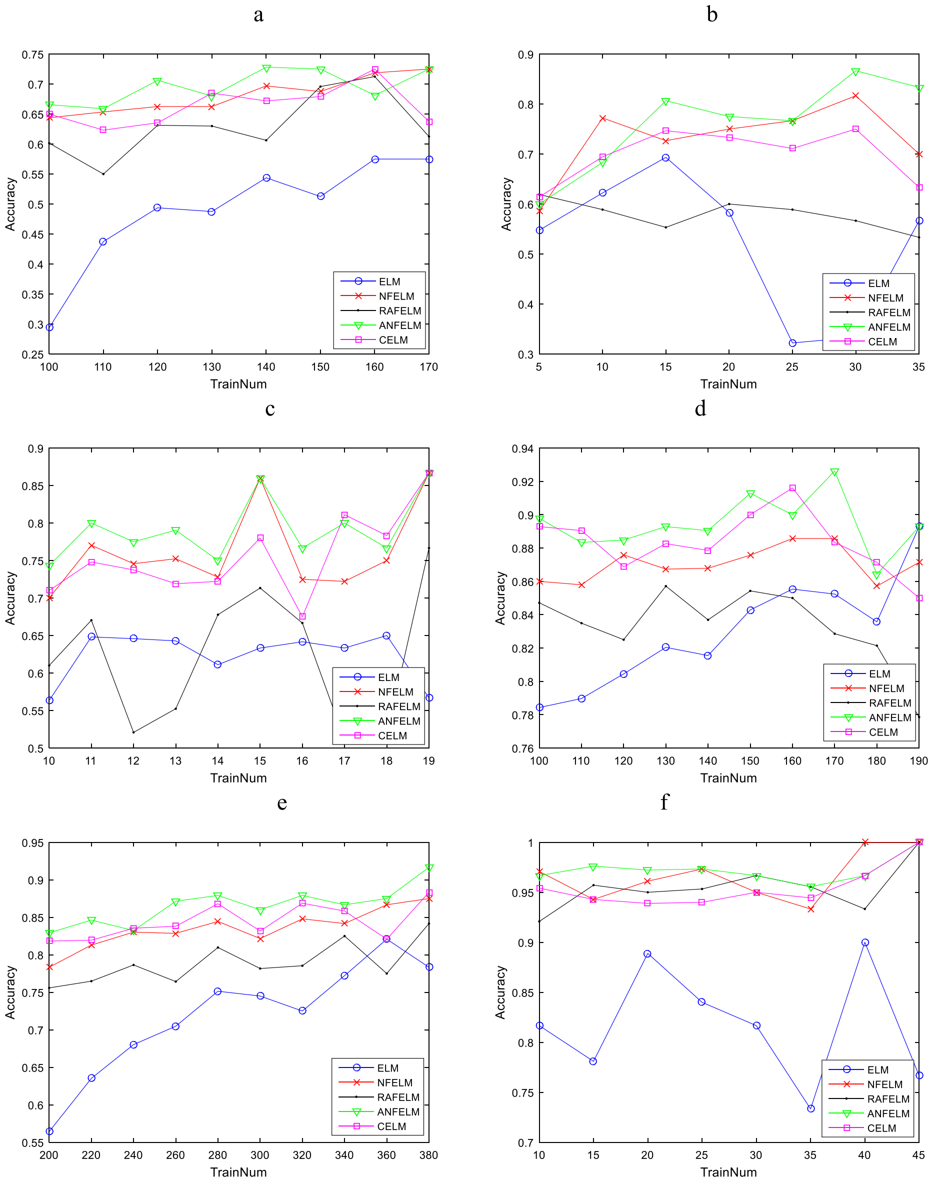


**Fig. 2** Different ELM algorithm identification curves. **a** vehicle; **b** wine; **c** movement; **d** segment; **e** waveform; **f** iris; **g** cmc; **h** pima; **i** diabetes; and **j** banknote

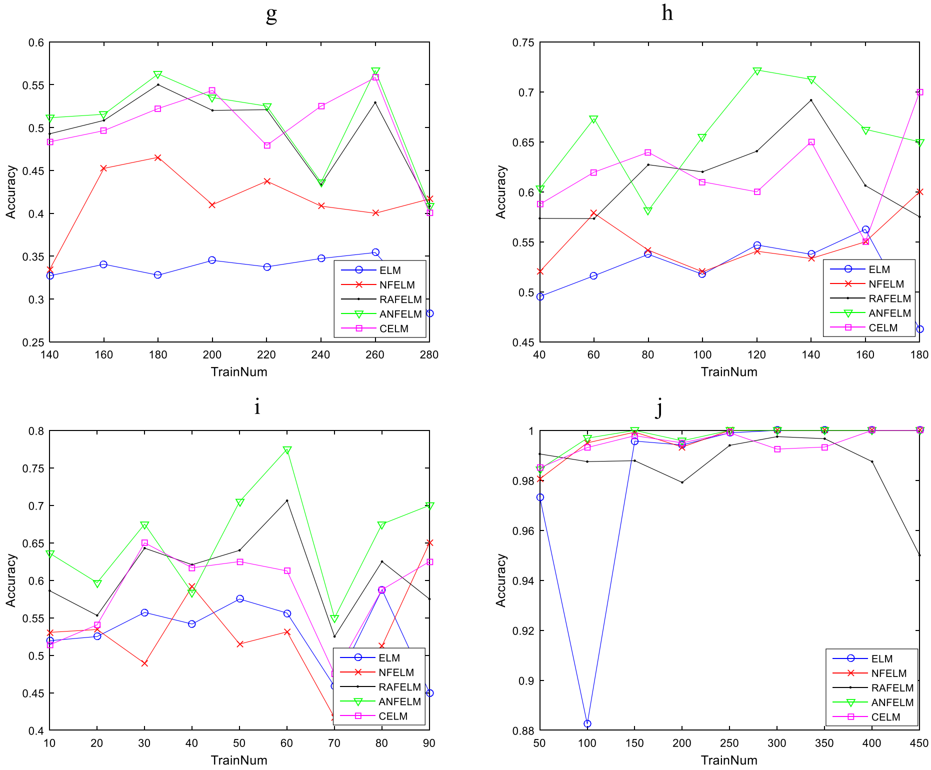


**Fig. 2** (continued)

The performance of ANFELM was compared to those of ELM, NFELM [38], RAFELM [18], and CELM [40]. The experiment was run 10 times with the same parameters and then averaged. The results are shown in Table 1.

In Table 1, there are different ELM algorithm accuracy rates of artificial datasets containing noise. The results in Table 1 show that the accuracy rate of the ANFELM algorithm is better than that of other algorithms; ANFELM has a good classification effect on noise-containing data, and its modifications enhance the robustness of the ELM algorithm to noise data.

## 5.2 Experiments with UCI benchmark datasets

In this section, we will experiment with ANFELM, ELM, NFELM [38], RAFELM [18], and CELM [40] using the UCI dataset [1]. The activation function of ANFELM, ELM, and NFELM [38], is sigmoid. RAFELM uses the improved activation function in [18]. The number of hidden layer nodes in the above five ELM algorithms is 500. For the fairness of the experiment, ELM and its improved algorithms all use the same penalty parameters (we try to find the optimal penalty parameters to make ELM and its improved algorithms obtain better results). In Fig. 2, the abscissa “TrainNum” indicates the number of training samples per class. In the experiment, we first randomly select a part of the data from each type of data sample as the experimental dataset for the experiment.

The descriptions of the datasets are shown in Table 2. The experimental results of using different algorithms on the UCI dataset are shown in Fig. 2 and Table 3.

**Table 2** UCI dataset description

| Datasets | Dim | Samples | Classes |
|---|---|---|---|
| vehicle | 18 | 846 | 4 |
| wine | 13 | 178 | 3 |
| movement | 90 | 360 | 15 |
| segment | 19 | 2310 | 7 |
| waveform | 40 | 5000 | 3 |
| iris | 4 | 150 | 3 |
| cmc | 9 | 1473 | 3 |
| diabetes | 8 | 768 | 2 |
| pima | 8 | 768 | 2 |
| banknote | 4 | 1372 | 2 |

### 5.2.1 Accuracy rate of different ELM algorithms using UCI datasets

Figure 2 shows the accuracy rate curves of different ELM algorithms generated with the UCI dataset. It can be seen from Fig. 2 that the accuracy rate curve of the proposed algorithm is higher than the accuracy rate curve of other algorithms. On the three datasets of `vehicle', `waveform', `diabetes', the accuracy of the proposed algorithm is significantly higher than that of ELM, NFELM, RAFELM and CELM with the increase of training samples. At the same time, the proposed algorithm shows good stability. When the number of training samples is 5 and 10 on the 'wine' data set, the accuracy of the proposed algorithm is lower than that of CELM algorithm. As the number of training samples increases gradually, the accuracy of the proposed algorithm is significantly higher than that of CELM and other algorithms.

Table 3 shows the accuracy rates of different ELM algorithms for the UCI dataset. Table 3 shows the average and maximum of the accuracy of Fig. 2. We add the accuracy of the different algorithms under different training samples in Fig. 2, and then get the mean as the experimental result of Table 3. From Table 3, we can see that the average and maximum values of the proposed algorithm in most datasets are significantly better than those of other algorithms. For our proposed method, the average accuracy of 'vehicle', 'wine' and 'movement' is 69.63%, 76.17% and 79.17%. The maximum accuracy on the two datasets of 'iris' and 'banknote' is 100%.

**Table 3** Maximum and average accuracy rates of the UCI dataset for different ELM algorithms (%)

| Dataset | ELM | | NFELM | | RAFELM | | ANFELM | | CELM | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Max | Average | Max | Average | Max | Average | Max | Average | Max | Average |
| vehicle | 57.50 | 48.99 | 72.50 | 68.13 | 71.25 | 63.00 | **72.81** | **69.63** | 72.50 | 66.34 |
| wine | 69.33 | 52.41 | 81.67 | 73.11 | 61.90 | 57.86 | **86.67** | **76.17** | 75.00 | 69.76 |
| movement | 65.00 | 62.36 | 86.67 | 76.20 | 76.67 | 62.06 | **86.67** | **79.17** | 86.67 | 75.53 |
| segment | 89.29 | 82.94 | 88.57 | 87.05 | 85.71 | 83.34 | **92.62** | **89.46** | 91.61 | 88.34 |
| waveform | 82.08 | 71.82 | 87.50 | 83.52 | 84.18 | 78.90 | **91.67** | **86.56** | 88.33 | 84.42 |
| iris | 90.00 | 81.79 | 100.00 | 96.64 | 100.00 | 95.46 | **100.00** | **97.22** | 100.00 | 95.46 |
| cmc | 35.42 | 33.28 | 46.53 | 41.56 | 55.00 | 49.53 | **56.67** | **50.76** | 55.83 | 50.10 |
| pima | 56.25 | 52.20 | 60.00 | 54.81 | 69.17 | 61.34 | **72.19** | **65.74** | 70.00 | 61.96 |
| diabetes | 58.75 | 53.00 | 65.00 | 53.02 | 70.63 | 60.82 | **77.50** | **65.51** | 65.00 | 58.29 |
| banknote | 100.00 | 98.27 | 100.00 | 99.65 | 99.75 | 98.57 | **100.00** | **99.75** | 100.00 | 99.51 |

In summary, the reason why the algorithm achieves good results is that, the ANFFELM algorithm takes into account the changes in the data geometry of the ELM feature map space when calculating the membership of the training samples, so the membership of the training samples is calculated in the feature mapping space of the data instead of in the data input space. NFELM takes into account the influence of noise and outliers on the ELM model and obtains a better accuracy effect, as

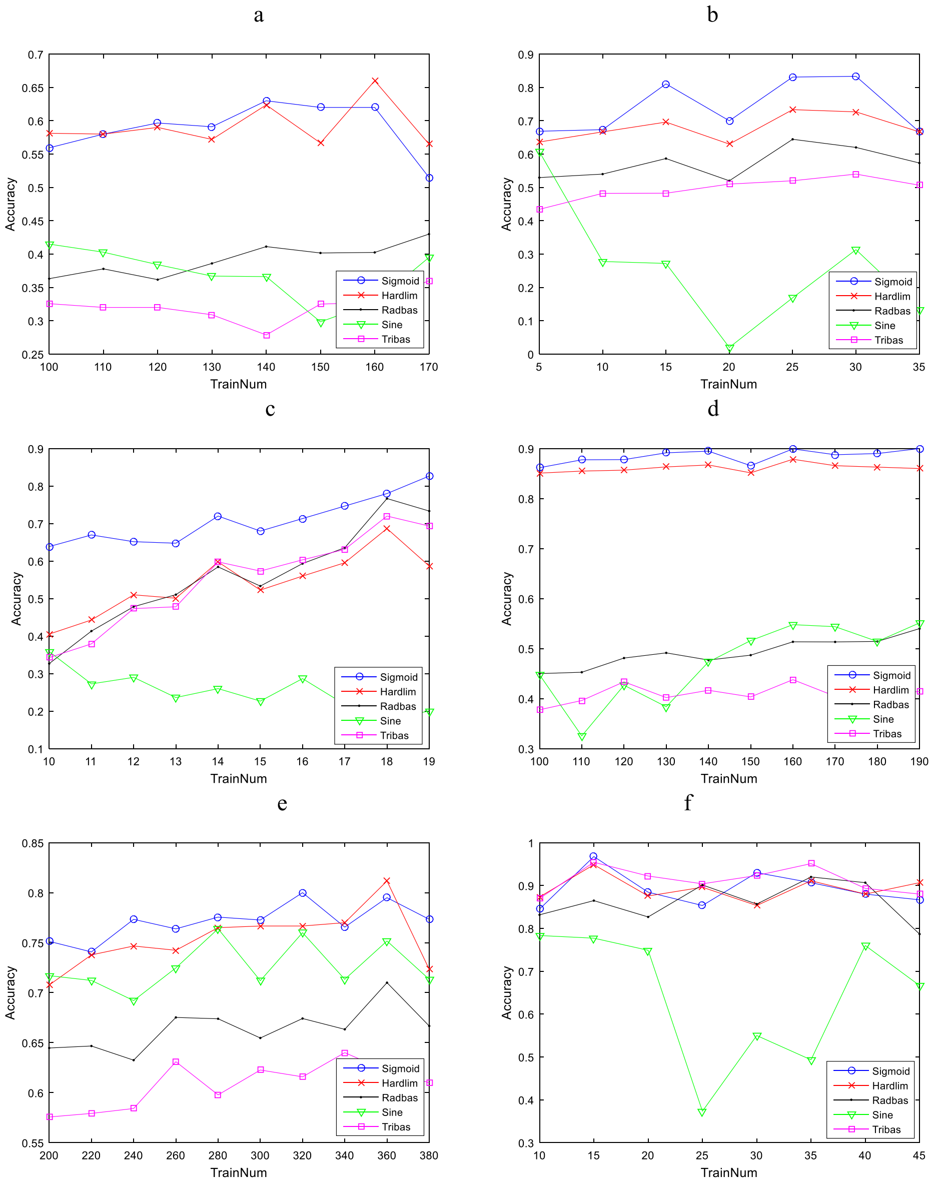


**Fig. 3** Different ELM algorithm identification curves. **a** vehicle; **b** wine; **c** movement; **d** segment; **e** waveform; **f** iris; **g** cmc; **h** pima; **i** diabetes; and **j** banknote

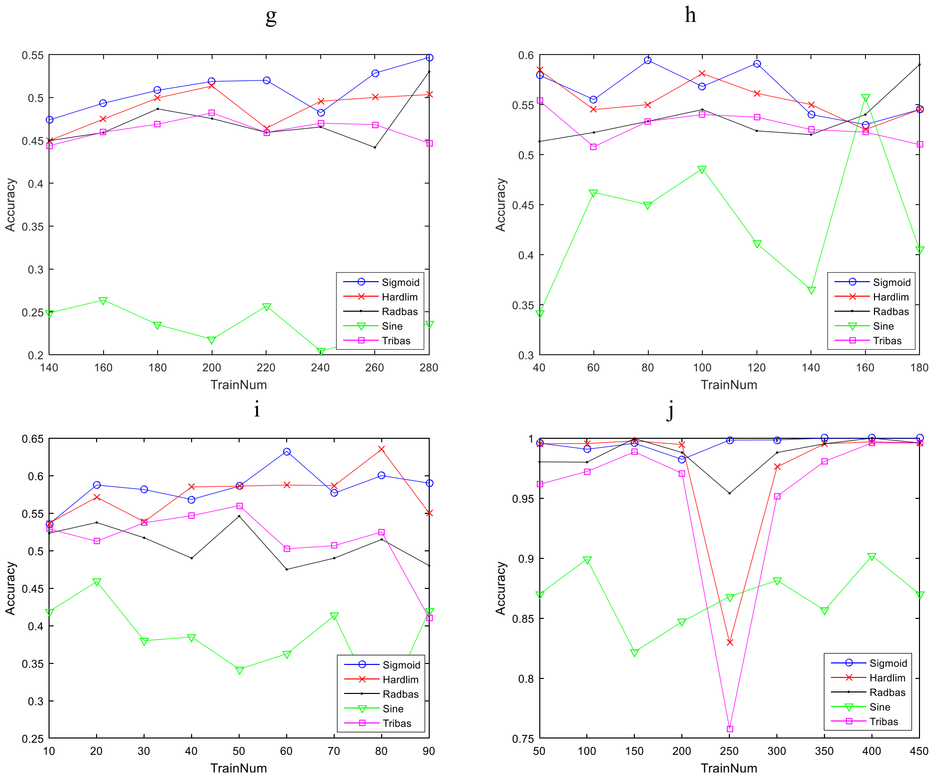


**Fig. 3** (continued)

derived from the accuracy rate curve. However, NFELM does not take into account the influence of ELM feature mapping space on data geometry. The overall accuracy rate of NFELM is lower than the accuracy rate of ANFELM. RAFELM improves the ELM algorithm from the perspective of the activation function and enhances the robustness of the activation function to noise and outliers. However, it does not eliminate the effects of noise and outliers on the ELM model. It can be seen from Fig. 2 that some accuracy rates of RAFELM on the datasets of wine, movement, segment are lower than the accuracy rate of ELM with an increasing number of training samples. With the increase in training samples, some accuracy rates of NFELM on the datasets pima and diabetes are lower than those of ELM. However, the accuracy rate curve resulting from the proposed algorithm and the UCI dataset is significantly higher than that of other algorithms. Table 3 shows the accuracy rate of different ELM algorithms for the UCI dataset. The table shows that the accuracy rate of ANFELM is significantly higher than the accuracy rate of other algorithms. Figure 2 and Table 3 show that the proposed algorithm is robust to noise and outliers.

### 5.2.2 The effect of the activation function on the performance of ANFELM

In this section, we study the effects of different activation functions on the proposed algorithm. We use the five activation functions sigmoid, hardlim, radbas, sine, and tribas in executing the ANFELM algorithm and perform experiments on the UCI dataset. The ANFELM algorithm selects the number of hidden layer nodes$L$ from$\{100, 200, 500, 800, 1000\}$. The penalty parameters$C$are selected in $\{2^{-5}, 2^{-4}, 2^{-3}, \ldots, 2^{10}\}$. In Fig. 3, the abscissa "TrainNum" indicates the number of training samples per class. In the experiment, we first randomly select a part of the data from each type of data sample as the experimental dataset. The experimental results are shown in Fig. 3 and Table 4.

Figure 3 and Table 4 show the accuracy rate of ANFELM for the UCI dataset using different activation functions. It can be seen from Fig. 3 that the accuracy rate of the sigmoid function on multiple datasets is better than that of other activation functions. The accuracy rate of the sine activation function is lower than the accuracy rate of the other activation functions on the datasets wine, movement, iris, cmc, pima, and diabetes. Hardlim's accuracy rate on the datasets vehicle, segment, waveform, iris, and cmc is very close to that of the sigmoid activation function. The accuracy rate of the radbas function and the tribas function is very close for multiple datasets. From Fig. 3, we also know that the accuracy rate of the sine activation function fluctuates greatly with increasing training samples in the three datasets iris, pima and diabetes. The tribas function has a large fluctuation in the accuracy rate curve when the number of training samples per class for the dataset banknote is 250. However, the accuracy rate curve of the activation function sigmoid is relatively stable and shows good properties. Table 4 shows the specific accuracy rate of the ANFELM algorithm for the UCI dataset using different activation functions. Table 4 shows that the accuracy rate of the sigmoid activation function is significantly better than that of other activation functions. Based on Fig. 3 and Table 4, the sigmoid activation function shows good robustness for the UCI dataset. Therefore, sigmoid is used as the activation function for this experiment.

### 5.2.3 Comparison of computational complexity of different ELM algorithms

To evaluate computational efficiency, we analysed the computing time complexity of ELM, RAFELM [18], NFELM [38], ANFELM, and CELM. In the equation $\beta = \left(HH^T + {}^{I}/_{C}\right)^{-1} H^T T$, $HH^T$ is an $L \times L$ matrix ($L$ is the number of hidden nodes). As in most cases, the number of hidden nodes, $L$, can be much smaller than the number of training samples, $N$. Namely, $L < < N$, and thus the computational cost is much lower in ELM than in LS-SVM and PSVM, which need to compute the inverse of a $N \times N$matrix [23]. The output weights of the algorithms RAFELM, CELM and ELM take the form $\beta = \left(HH^T + {}^{I}/_{C}\right)^{-1} H^T T$, while those of NFELM and ANFELM take the form $\beta = H^T \left(HH^T + {}^{I}/_{C}\right)^{-1} T$. All of the algorithms need to calculate the$L \times L$inverse matrix of $HH^T$, so the time complexity of ELM, RAFELM, MCVELM, GELM, and IELM is$O(L^3)$.

Table 5 shows the experimentally determined computational efficiencies of the compared algorithms. The experiments are conducted on a computer with an Intel(R) Core(TM) 3.40 GHZ CPU, 8 GB of RAM, and MATLAB 2015b.

Table 5 shows the running time of different ELM algorithms on the UCI dataset. Table 5 shows that the running time of the NFELM and ANFELM algorithms is higher than the running time of ELM. The reason is that the NFELM and ANFELM algorithms use a clustering algorithm to calculate the membership degree, which takes a certain time. Table 5 shows that the running time of the ANFELM algorithm is higher than that of the NFELM algorithm. The reason is that the ANFELM algorithm uses clustering and K-nearest neighbour

Table 4 The UCI dataset maximum and average accuracy rate for ANFELM using different activation functions (%)

| Dataset | Sigmoid | | Hardlim | | Radbas | | Sine | | Tribas | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Max | Average | Max | Average | Max | Average | Max | Average | Max | Average |
| vehicle | 63.00 | 58.90 | **66.00** | **59.23** | 43.00 | 39.18 | 41.50 | 36.95 | 36.00 | 32.07 |
| wine | **83.33** | **74.05** | 73.33 | 67.94 | 64.44 | 57.34 | 60.76 | 25.61 | 54.00 | 49.66 |
| movement | **82.67** | **70.74** | 68.67 | 54.09 | 76.67 | 55.75 | 35.87 | 25.29 | 72.00 | 54.92 |
| segment | **90.00** | **88.46** | 87.86 | 86.12 | 54.00 | 49.20 | 55.14 | 47.30 | 43.79 | 40.95 |
| waveform | 80.00 | **77.11** | **81.17** | 75.38 | 71.00 | 66.42 | 76.39 | 72.60 | 64.00 | 60.78 |
| iris | **96.76** | 89.17 | 94.86 | 89.31 | 92.20 | 86.18 | 78.33 | 72.60 | 95.43 | **91.23** |
| cmc | **54.67** | **50.89** | 51.33 | 48.75 | 53.00 | 47.09 | 26.38 | 23.54 | 48.20 | 46.23 |
| pima | **59.42** | **56.29** | 58.44 | 55.52 | 59.00 | 53.59 | 55.75 | 43.48 | 55.38 | 52.88 |
| diabetes | 63.25 | **58.42** | **63.50** | 57.52 | 54.60 | 50.82 | 45.88 | 38.45 | 56.00 | 51.44 |
| banknote | **100.00** | **99.57** | 99.77 | 97.41 | 100.00 | 98.68 | 90.20 | 86.85 | 99.60 | 95.28 |

methods to calculate the membership degree. Therefore, its running time is longer than that of the NFELM algorithm. Both the RAFELM and CELM algorithm run faster than ELM on the movement, segment, cmc, pima, and diabetes datasets. This difference may be caused by their different data optimization forms in data operations. Based on Tables 1, 3, 5, 6 and 7, we analyse the performance of the proposed algorithm ANFELM in terms of its classification accuracy rate and time efficiency. From Tables 1, 3, 6 and 7, we can see that ANFELM has good classification ability and is superior to other ELM algorithms. From Table 5, ANFELM does not have much advantage in time overhead, speed of classification and classification performance based on ELM. ANFELM can be used as an effective classifier in pattern accuracy.

## 5.3 Experiments with text data

In recent years, researchers have applied the ELM algorithm to text classification tasks [39]. In this section, we apply the algorithm proposed in this paper to test the validity of a text dataset verification algorithm. We choose the Reuters-21,578 dataset and the 20newsgroup dataset as experimental datasets and experimented with ANFELM, SVM, ELM, and RELM. The experimental results are shown in Tables 6 and 7.

Table 5 Running time of different ELM algorithms on UCI datasets (Training time + Testing time)(s)

| Dataset | ELM | NFELM | RAFELM | ANFELM | CELM |
|---|---|---|---|---|---|
| vehicle | 0.1875 | 2.7969 | 0.3516 | 3.6797 | 0.2500 |
| wine | 0.0703 | 0.2422 | 0.1172 | 0.3438 | 0.2813 |
| movement | 0.5547 | 0.6797 | 0.2422 | 2.7266 | 0.2891 |
| segment | 0.7188 | 20.3672 | 0.8828 | 23.8750 | 0.5938 |
| waveform | 0.4609 | 11.0000 | 0.7813 | 13.0625 | 0.4688 |
| iris | 0.2109 | 0.2109 | 0.2344 | 0.5899 | 0.2656 |
| cmc | 0.3281 | 3.7500 | 0.3750 | 4.9063 | 0.1875 |
| pima | 0.2656 | 0.6094 | 0.1797 | 0.9844 | 0.1641 |
| diabetes | 0.1875 | 0.3594 | 0.1094 | 0.5625 | 0.1172 |
| banknote | 0.2969 | 3.2500 | 0.4922 | 3.9375 | 0.3203 |

**Table 6** Experimental results of using different ELM algorithms on Reuters-top10 (%)

| Category | P(%) | | | | R(%) | | | | F1(%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ELM | SVM | RELM] | ANFELM | ELM | SVM | RELM | ANFELM | ELM | SVM | RELM | ANFELM |
| earn | 95.67 | **98.66** | 97.25 | 98.56 | 97.69 | 98.75 | 98.75 | **98.94** | 96.67 | 98.70 | 98.00 | **98.75** |
| acq | 93.47 | 94.68 | 95.69 | **96.02** | 94.68 | **97.58** | 96.77 | 97.26 | 94.07 | 96.11 | 96.23 | **96.63** |
| crude | 93.26 | 95.56 | 96.77 | 97.78 | 84.69 | 87.76 | **91.84** | 89.80 | 88.77 | 91.49 | 94.24 | **93.62** |
| trade | 91.55 | **94.67** | 93.24 | 88.89 | 89.04 | 97.26 | 94.52 | **98.63** | 90.28 | **95.59** | 93.88 | 93.51 |
| money-fx | 79.66 | 83.78 | **85.78** | 85.71 | 68.12 | **89.86** | 76.81 | 89.69 | 73.44 | **86.71** | 80.92 | 86.33 |
| interest | 77.36 | 93.48 | 86.00 | **95.65** | 71.93 | 75.44 | 75.44 | **77.19** | 74.55 | 83.50 | 80.37 | **85.54** |
| ship | 84.85 | 87.10 | 93.55 | **88.24** | 80.00 | 77.14 | 82.86 | **85.71** | 82.35 | 81.82 | **87.88** | 86.96 |
| sugar | 100.00 | 100.00 | 100.00 | **100.00** | 91.67 | 83.33 | 95.83 | **95.83** | 95.65 | 90.91 | 97.87 | **97.87** |
| coffee | 94.95 | 95.24 | 95.24 | **95.45** | 100.00 | 95.24 | 95.24 | **100.00** | 97.67 | 95.24 | 95.24 | **97.67** |
| gold | 94.44 | 95.00 | **100.00** | 94.74 | 85.00 | 95.00 | **100.00** | 90.00 | 89.47 | 95.00 | **100.00** | 92.31 |
| Average | 90.57 | 93.82 | **94.32** | 94.10 | 86.28 | 89.74 | 90.81 | **92.03** | 88.29 | 91.54 | 92.46 | **92.91** |

**Table 7** Experimental results of using different ELM algorithms on 20newsgroup –top7 (%)

| Category | P(%) | | | | R(%) | | | | F1(%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ELM | SVM | RELM | ANFELM | ELM | SVM | RELM | ANFELM | ELM | SVM | RELM | ANFELM |
| Alt | 79.66 | 85.02 | 81.25 | **84.86** | 58.93 | 60.50 | 61.13 | **66.77** | 67.75 | 70.70 | 69.77 | **74.74** |
| Computers | 81.31 | 87.50 | 81.41 | **87.95** | 89.67 | **94.17** | 90.95 | 93.71 | 85.28 | 90.71 | 85.91 | **90.74** |
| Miscellaneous | 88.12 | **88.53** | 86.06 | 84.55 | 68.46 | 77.18 | 72.82 | **77.18** | 77.06 | **82.47** | 78.89 | 80.70 |
| Recreation | 86.37 | 92.36 | 86.18 | **93.59** | 89.69 | 95.79 | 89.06 | **96.48** | 88.00 | 94.04 | 87.60 | **95.01** |
| Science | 80.41 | 86.96 | 80.23 | **88.24** | 77.20 | 84.48 | 76.31 | **85.56** | 78.77 | 85.71 | 78.22 | **86.88** |
| Social | 79.14 | **83.98** | 83.20 | 83.17 | 74.37 | 81.66 | 75.88 | **85.68** | 76.68 | 82.80 | 79.37 | **84.41** |
| Talk | 82.41 | 88.54 | 83.94 | **91.20** | 82.09 | 87.86 | 81.94 | **88.47** | 82.35 | 88.19 | 82.92 | **89.82** |
| Average | 82.49 | 87.56 | 83.18 | **87.65** | 77.20 | 83.09 | 78.30 | **84.84** | 79.00 | 84.95 | 80.38 | **86.04** |

### 5.3.1 Text datasets

Reuters[1] dataset: Reuters is a textual dataset consisting of 21,578 financial news texts published by Reuters. These news texts are divided into 5 broad categories and 135 subcategories. In this experiment, a text dataset consisting of the ten most commonly used subcategories was selected. After data preprocessing, the dataset contains 2201 term features (using TFIDF to calculate weights), 10 text categories, 5228 training set texts, and 2057 test set texts.

20newsgroup[1] dataset: 20newsgroup is a collection of approximately 20,000 news texts that are divided into 20 different groups, one for each topic. The dataset is divided into a training set and a test set using a standard "ModApte" partition. After data preprocessing, the dataset contains 27,808 feature words, 20 categories transformed into 7 text categories, 11,314 training set samples, and 7532 test set samples. In the experiment, we selected 2260 term features for the experiment.

### 5.3.2 Evaluation measures

The experiment uses the accuracy rate (P), recall rate (R), and$F_1$metrics to evaluate the effectiveness of the classification algorithm.

For a dataset: The number of samples is represented by$N$, where the number of positive samples is represented as$N_P$and the number of negative samples is represented as $N_L$; then $N_P = N_{TP} + N_{FN}$, $N_{TP}$ is actually a positive sample, and the prediction is also a positive sample.$N_{FN}$is a positive sample, but the prediction is a negative sample $N_L = N_{TN} + N_{FP}$, $N_{TN}$ is a negative sample, but the prediction is a positive sample.$N_{FP}$is actually a negative sample, but the prediction is a positive sample. The accuracy, recall, and $F_1$ metrics are as follows:

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{36}$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}} = \frac{N_{TP}}{N_P} \tag{37}$$

$$F_1 = \frac{2P*R}{P+R} \tag{38}$$

Tables 6 and 7 show the experimental results of using the ELM, SVM, RELM [1] and ANFELM algorithms on Reuters-top10 and 20newsgroup –top7. The accuracy rate, recall rate and F value of each data category were used to evaluate the classification effects of different algorithms. At the same time, the average accuracy, average recall rate and average F value were given under the overall category. The experimental results in Tables 6 and 7 show that the classification results of the ANFELM algorithm using the three evaluation indicators are better than those of the other algorithms. The SVM classification results in multiple categories using the Reuters-top10 dataset are better than the results from the ELM algorithm. The RELM algorithm classification is better than the SVM algorithm classification. The reason for these differences is that the RELM [1] algorithm adds regular terms to the ELM algorithm

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

to enhance the ELM algorithm generalization capability. Table 7 shows that the classification metrics of the ELM and RELM algorithms are lower than the classification metrics of the SVM algorithm when they are performed on the 20newsgroup –top7 dataset. However, the classification results of the ANFELM algorithm are better than those of the SVM on the same dataset. This result suggests that ANFELM is also an effective method for text categorization.

## 6 Conclusion

In this paper, to overcome the existing ELM algorithm's issues in dealing with noise and outliers, clustering and K-nearest neighbour methods are used to calculate the membership of each sample, and the membership is introduced into the ELM model. It is worth mentioning that the ANFELM algorithm takes into account the change in the data geometry of the ELM feature mapping space when calculating the membership of the training sample. Therefore, the membership of the training sample is calculated in the feature mapping space of the data instead of the data input space. The geometric relationship of the sample is better characterized in ANFELM than in ELM. The robustness of the ELM algorithm in processing noise and outliers is enhanced. Experiments on 20 UCI benchmark datasets indicate that ANFELM outperforms other methods in terms of one evaluation criteria, classification accuracy (ACC). The proposed ANFELM method was also validated on text categorization tasks, on which it achieves superior performance to other baselines.

**Acknowledgements** This work is partially supported by grant from the Natural Science Foundation of China (No. 61632011, 61572102, 61702080, 61602079) and the Fundamental Research Funds for the Central Universities (NO. DUT18ZD102, DUT17RC(3)016).

## References

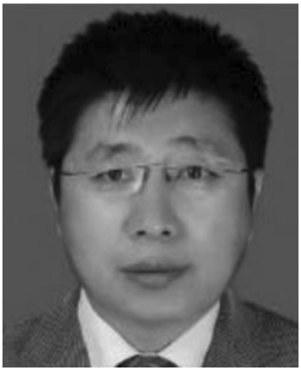
1. Bache K, Lichman M (2013) UCI machine learning repository. In: School Inf. Comput. Sci., Univ. California, Irvine, CA, USA. Available: http://archive.ics.uci.edu/ml
2. Barro S, Ribeiro J (2014) Direct kernel perceptron (DKP): ultra-fast kernel ELM-based classification with noniterative closed-form weight calculation. Neural Netw 50(2):60–71
3. Cao FX, Yang ZJ, Ren JC, Jiang MY, Ling WK (2017) Linear vs. nonlinear extreme learning machine for spectral-spatial classification of hyperspectral images. Sensors. https://doi.org/10.3390/s17112603
4. Castaño A, Fernández-Navarro F, Hervás-Martínez C (2013) PCA-ELM: a robust and pruned extreme learning machine approach based on principal component analysis. Neural Process Lett 37(3):377–392
5. Gu Y, Chen YQ, Liu JF, Jiang XL (2015) Semi-supervised deep extreme learning machine for Wi-Fi based localization. Neurocomputing 166(C):282–293
6. Huang GB (2014) An insight into extreme learning machines: random neurons, random features and kernels. Cogn Comput 6(3):376–390
7. Huang G, Song SJ (2014) Semi-supervised and unsupervised extreme learning machines. IEEE Trans Cybern 44(12):2405–2417
8. Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of international joint conference on neural networks, IJCNN2004, Budapest, Hungary, 2, pp 985–990
9. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
10. Huang GB, Chen L, Siew CK (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17(4):879–892

11. Huang GB, Zhu QY, Mao KZ, Siew CK, Saratchandran P, Sundararajan N (2006) Can threshold networks be trained directly? IEEE Trans Circuits Syst Express Briefs 53(3):187–191
12. Huang G-B, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B Cybern 42(2):513–529
13. Jiang XF, Yi Z, Lv JC (2006) Fuzzy SVM with a new fuzzy membership function. Neural Comput & Applic 15(3–4):268–276
14. Lavneet S, Girija C, Dharmendra S (2012) A novel approach to protein structure prediction using PCA based extreme learning machines and multiple kernels. In: International conference on algorithms and architectures for parallel processing, pp 292–299
15. Li L, Wang CY, Li W, Chen JB (2018) Hyperspectral image classification by AdaBoost weighted composite kernel extreme learning machines. Neurocomputing 275:1725–1733
16. Lin S, Liu X, Fang J, Xu Z (2015) Is extreme learning machine feasible? A theoretical assessment (part II). IEEE Trans Neural Netw Learn Syst 26(1):21–34
17. Liu XY, Gao CH, Li P (2012) A comparative analysis of support vector machines and extreme learning machines. Neural Netw 33(9):58–66
18. Liu S, Feng L, Xiao Y (2014) Robust activation function and its application: semi-supervised kernel extreme learning method. Neurocomputing 144(1):318–328
19. Liu X, Lin S, Fang J, Xu Z (2015) Is extreme learning machine feasible? a theoretical assessment (part I). IEEE Trans Neural Netw Learn Syst 26(1):7–20
20. Liu B, Xia SX, Meng FR, Zhou Y (2016) Manifold regularized extreme learning machine. Neural Comput & Applic 27(2):255–269
21. Lv F, Han M, Qiu T (2017) Remote sensing image classification based on ensemble extreme learning machine with stacked autoencoder. IEEE Access 5(99):1725–1733
22. Ma YP, Niu PF, Yan SS, Li GQ (2018) A modified online sequential extreme learning machine for building circulation fluidized bed boiler's NOx emission model. Appl Math Comput 334:214–226
23. Raghuwanshi BS, Shukla S (2018) Class-specific extreme learning machine for handling binary class imbalance problem. Neural Netw 105:206–217
24. Raghuwanshi BS, Shukla S (2018) UnderBagging based reduced Kernelized weighted extreme learning machine for class imbalance learning. Eng Appl Artif Intell 74:252–270
25. Rumelhart D, Hinton G, Williams R (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536
26. Sahani M, Dash PK (2018) Variational mode decomposition and weighted online sequential extreme learning machine for power quality event patterns accuracy. Neurocomputing 310:10–27
27. Su H, Cai Y, Du Q (2017) Firefly-algorithm-inspired framework with band selection and extreme learning machine for hyperspectral image classification. IEEE J Sel Top Appl Earth Obs Remote Sens 10(1):309–320
28. Tang XL, Han M (2010) Ternary reversible extreme learning machines: The incremental tri-training method for semi-supervised classification. Knowl Inf Syst 23(3):345–372
29. Vapnik VN (1995) The nature of statistical learning theory. Springer 8(6):988–999
30. Wang XZ, Shao QY, Miao Q, Zhai JH (2013) Architecture selection for networks trained with extreme learning machine using localized generalization error model. Neurocomputing 102(2):3–9
31. Wang Q, Wang WG, Nian R, He B (2016) Manifold learning in local tangent space via extreme learning machine. Neurocomputing 174(PA):18–30
32. Xia SX, Meng FR, Liu B, Zhou Y (2015) A kernel clustering-based possibilistic fuzzy extreme learning machine for class imbalance learning. Cogn Comput 7(1):74–85
33. Xiao WD, Zhang J, Li YJ, Zhang S, Yang WD (2017) Class-specific cost regulation extreme learning machine for imbalanced classification. Neurocomputing 261:70–82
34. Yu Q, Miche Y, Eirola E, van Heeswijk M, Severin E, Lendasse A (2013) Regularized extreme learning machine for regression with missing data. Neurocomputing 102(2):45–51
35. Zhang W, Ji H (2013) Fuzzy extreme learning machine for classification. Electron Lett 49(7):448–450
36. Zhang K, Luo MX (2015) Outlier-robust extreme learning machine for regression problems. Neurocomputing 151:1519–1527
37. Zhang R, Lan Y, Huang GB, Xu ZB (2012) Universal approximation of extreme learning machine with adaptive growth of hidden nodes. IEEE Trans Neural Netw Learn Syst 23(2):365–371
38. Zheng EH, Liu JY (2013) A new fuzzy extreme learning machine for regression problems with outliers or noises, 9th international conference, ADMA, pp 524–534
39. Zheng WB, Qian YT, Lu HJ (2013) Text categorization based on regularization extreme learning machine. Neural Comput & Applic 22(3-4):447–456

40. Zhu WT, Miao J, Qing LY (2014) Constrained extreme learning machine: a novel highly discriminative random feedforward neural network. In: International joint conference on neural networks, pp 6–11
41. Zou QY, Wang XJ, Zhou CJ, Zhang Q (2018) The memory degradation based online sequential extreme learning machine. Neurocomputing 275:2864–2879

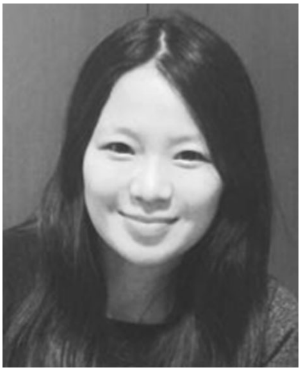
**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Yonghe Chu** received the M.S. degree in College of Computer and Information Technology, Liaoning Normal University, China, in 2017. Currently, he is working toward the Ph.D. degree in the School of ComputerScience and Technology, Dalian University of Technology, China. His research interests include pattern, recognition computer vision and machine learning.

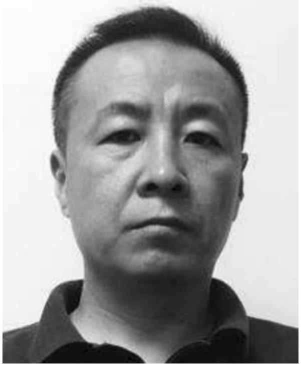


**Hongfei Lin** received the M.S. degree from the Dalian University of Technology, Dalian, China, in 1992 and the Ph.D.degree from Northeastern University, Shenyang, China, in 2000. He is currently a Professor with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology. He has authored over 100 scientific papers in various journals, conferences, and books. His research projects are funded by Natural Science. His current research interests include text mining for biomedical literature, biomedical hypothesis generation, information extraction from huge biomedical resources, learning to rank, sentimental analysis, and opinion mining.

**Liang Yang** received the M.S. degree from the Dalian University of Technology, Dalian, China, and the Ph.D.degree from the Dalian University of Technology, Dalian, China, He is currently a instructor with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology. His current research interests include sentimental analysis, and opinion mining.



**Dongyu Zhang** is working toward the Ph.D. degree in the School of ComputerScience and Technology, Dalian University of Technology, China. Her research interests include sentimental analysis, and opinion mining.



**Shaowu Zhang** is a Professor with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology. His current research interests include sentimental analysis, and opinion mining.

**Yufeng Diao** is working toward the Ph.D. degree in the School of ComputerScience and Technology, Dalian University of Technology, China. Her research interests include sentimental analysis, and opinion mining.

**Deqin Yan** is professor at College of Computer and Information Technology, Liaoning Normal University. He received Ph.D. at Nankai University in 1999. His research interest is pattern recognition.