TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 12/15 pp691-701
Volume 22, Number 6, December 2017

A Heterogeneous Ensemble of Extreme Learning Machines with Correntropy and Negative Correlation

Adnan O. M. Abuassba, Yao Zhang, Xiong Luo*, Dezheng Zhang, and Wulamu Aziguli*

Abstract: The Extreme Learning Machine (ELM) is an effective learning algorithm for a Single-Layer Feedforward Network (SLFN). It performs well in managing some problems due to its fast learning speed. However, in practical applications, its performance might be affected by the noise in the training data. To tackle the noise issue, we propose a novel heterogeneous ensemble of ELMs in this article. Specifically, the correntropy is used to achieve insensitive performance to outliers, while implementing Negative Correlation Learning (NCL) to enhance diversity among the ensemble. The proposed Heterogeneous Ensemble of ELMs (HE^{2}LM) for classification has different ELM algorithms including the Regularized ELM (RELM), the Kernel ELM (KELM), and the L_2-norm-optimized ELM (ELML2). The ensemble is constructed by training a randomly selected ELM classifier on a subset of the training data selected through random resampling. Then, the class label of unseen data is predicted using a maximum weighted sum approach. After splitting the training data into subsets, the proposed HE^{2}LM is tested through classification and regression tasks on real-world benchmark datasets and synthetic datasets. Hence, the simulation results show that compared with other algorithms, our proposed method can achieve higher prediction accuracy, better generalization, and less sensitivity to outliers.

Key words: Extreme Learning Machine (ELM); ensemble; classification; correntropy; negative correlation

1 Introduction

Ensemble learning is a machine learning paradigm used to enhance performance results[1]. Ensembles are known as a mixture of experts to reduce overfitting and errors from all combined base learners and have proved their performance in many real-world applications[2]. To improve the accuracy and stability of the ensemble, different techniques have been developed. These techniques vary by the training data used, the type of algorithms used, and the combination methods that are followed. Bagging[3], boosting[4], and their variants, such as Adaboost[5], are some of the popular ensemble techniques. Generally, traditional Neural Networks (NNs) suffer from overfitting and local optimum issues, and have remained an active research subject for performance improvement by different methods[6, 7]. Then, the Extreme Learning Machine (ELM) for NNs is effective for solving many problems, such as classification and regression[8, 9]. It has good theoretical support and it performs well in practical applications[10, 11].

Even though ELM has reliable performance, there is still a lot of room for improvement[12]. To improve the accuracy and generalization, some modifications have been recently introduced on the basis of the ELM,

• Adnan O. M. Abuassba, Xiong Luo, Dezheng Zhang, and Wulamu Aziguli are with the School of Computer and Communication Engineering, University of Science and Technology Beijing (USTB), Beijing 100083, China, and the Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China. E-mail: adnanasba@yahoo.com; xluo@ustb.edu.cn; zdzchina@126.com; aziguli@ustb.edu.cn.
• Yao Zhang is with the Tandon School of Engineering, New York University, Brooklyn, NY 11201, USA. E-mail: yz4567@nyu.edu.
* To whom correspondence should be addressed.
Manuscript received: 2016-10-29; revised: 2017-03-05; accepted: 2017-03-07

such as the Optimally Pruned ELM (OP-ELM)[8], L_2-norm-optimized ELM (ELML2)[9], regularized ELM (RELM)[13], Kernel ELM (KELM)[14], and many others[15, 16].

Furthermore, ensemble learning is a cheap alternative due to its optimization performance. Accordingly, several approaches were proposed to generate ensemble based ELM, such as DELM[17] and EnELM[18]. Some of the proposed ELM ensembles were successful in achieving reliable performance for classification of hyperspectral image[19]. The Bagging-ELM (B-ELM)[20] is another ELM ensemble classifier, which leverages the bag of little bootstraps technique and is found efficient for large-scale data classification. An Online Sequential-ELM (OS-ELM) based framework supports ensemble methods including Bagging and subspace partitioning[21]. Due to ELM's high performance, ELM ensembles were employed in many real-world applications. Here we mention just a few examples of those applications since there are numerous examples. A landmark recognition method was proposed using the ELM ensemble and feature selection technique[22]. The face-matching based ELM ensemble was proposed[23]. An ELM ensemble based on the Min-Max-Modular network was proposed in big data applications[24]. An heterogeneous ELM ensemble was designed in addressing classification problems[25].

Meanwhile, to improve the performance of the ensemble, the Negative Correlation Learning (NCL) technique is developed, which constructs base classifiers based on negative correlation between the learner and the other classifiers in the ensemble, resulting in a diverse and accurate model. Diversity among the performance of each learner in the ensemble is essential for combining the predictions from several member classifiers. Different techniques are followed to introduce diversity among member classifiers. For example, a cross-validation was proposed[17, 18, 26]. An ensemble learning for NNs via negative correlation was first proposed[27]. In addition, the correntropy is used as a measure to achieve insensitive performance to outliers. Then, a regularized correntropy was used to train the ELM and a novel algorithm, namely ELM-RCC, was proposed[28].

Generally, the proposed ELM ensembles only have one base classifier algorithm for the training members and can be considered as homogenous ensemble models[29, 30]. Moreover, the correntropy is only used for learning the base classifiers. A homogeneous ensemble model to train the classifiers was proposed[17, 18, 26]. This article proposes a heterogeneous ensemble of ELMs using correntropy and NCL, which combines both data levels and algorithmic levels in the ensemble learning model.

Since correntropy can improve the anti-noise ability of the ELM, and NCL can enhance ensemble performance, we propose a novel scheme that combines them into one ensemble learning framework. Furthermore, in the heterogeneous ensemble model, different ELM algorithms are integrated for member training. Specifically, three types of ELM algorithms, namely, ELML2[9], RELM[13], and KELM[14], are used. These learners are chosen based on their better generalization, regularization, and resilience to the outliers. Moreover, a random resampling strategy is developed to split the training data into subsets. Each member classifier is learned on a randomly chosen data subset through a randomly selected base ELM algorithm.

The rest of this article is organized as follows. Descriptions of correntropy, NCL, and the base learners of our scheme are briefly reviewed in Section 2. The HE^{2}LM scheme and implementation details are presented in Section 3. The weighting method is employed for combining the predictions of all members in the HE^{2}LM. Simulation results of the proposed HE^{2}LM are provided in Section 4 while comparing its performance with the RELM and the ELM-RCC algorithms.

2 Background

2.1 ELM theory

According to the ELM theorem, it is implemented with random hidden nodes. Let $(\boldsymbol{x}_j, \boldsymbol{t}_j)_{j=1}^N$ be the input for training, where $\boldsymbol{x}_j$ represents the training data vector, $\boldsymbol{t}_j$ represents the training data target, and N represents the number of input data. The ELM aims to minimize the output weights β and the Mean Square Error (MSE) simultaneously, as follows[9, 31]:

$$f_{\text{ELM}} = \|\beta\|_p^{\sigma_1} + \lambda \|\boldsymbol{H}\beta - \boldsymbol{T}\|_q^{\sigma_2} \tag{1}$$

where $\sigma_1, \sigma_2 > 0$, $\lambda > 0$, $p, q = \frac{1}{2}, 1, 2, \cdots, \infty$, and $\boldsymbol{H}$ is the hidden layer output matrix defined by

$$\boldsymbol{H} = \begin{bmatrix} \hbar(\boldsymbol{x_1}) \\ \vdots \\ \hbar(\boldsymbol{x}_N) \end{bmatrix} = \begin{bmatrix} h(\boldsymbol{x}_1) & \cdots & h_L(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ h(\boldsymbol{x}_N) & \cdots & h_L(\boldsymbol{x}_N) \end{bmatrix} \tag{2}$$

where L denotes the number of hidden nodes, and for

input vector $\boldsymbol{x}_j$, $\hbar(\boldsymbol{x}_j) = [h_i(\boldsymbol{x}_j)]_{i=1}^{L}$ represents the output vector in the hidden layer. Furthermore, $\boldsymbol{T}$ is the desired result of the input data, defined as

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{t}_N^{\mathrm{T}} \end{bmatrix} \tag{3}$$

The ELM training algorithm is summarized by three steps[9]:

- Set the biases b_j and the input weights a_j in a random manner;
- Compute the matrix $\boldsymbol{H}$;
- Compute β.

Here, β is obtained by

$$\beta = \boldsymbol{H}^{\dagger}\boldsymbol{T} \tag{4}$$

where $\boldsymbol{H}^{\dagger}$ represents the Moore-Penrose (MP) inverse. The MP inverse is computed by applying the orthogonal projection: $\boldsymbol{H}^{\dagger} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}$, given that $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$ is nonsingular; or $\boldsymbol{H}^{\dagger} = \boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})^{-1}$, given that $\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$ is nonsingular. In accordance with the ridge regression theory, a positive matrix $\boldsymbol{I}/\lambda$ is added to the $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$ or $\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$. Then, we have a solution which is equivalent to the optimized ELM with $\sigma_1 = \sigma_2 = 2$[9]. Hence, we can have

$$\beta = \boldsymbol{H}^{\mathrm{T}}\left(\frac{\boldsymbol{I}}{\lambda} + \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\right)^{-1}\boldsymbol{T} \tag{5}$$

$$g(\boldsymbol{x}) = h(\boldsymbol{x})\beta = h(\boldsymbol{x})\boldsymbol{H}^{\mathrm{T}}\left(\frac{\boldsymbol{I}}{\lambda} + \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\right)^{-1}\boldsymbol{T} \tag{6}$$

Considering the advantages of the ELM, we propose to use it in our ensemble to achieve better classification results. In ensemble learning, we use three types of ELM versions to improve diversity among the base classifiers. Overall, the proposed ensemble is designed to enhance performance and it is less sensitive to noise.

2.2 Base classifiers

Three types of ELM classifiers, i.e., RELM, KELM, and ELML2, are used as base classifiers to construct the HE^{2}LM ensemble. Here, we briefly introduce the features of the selected base ELM classifiers.

First, RELM is a constrained and optimized version of the ELM for regression and multiclass classification[13]. The RELM provides a good tradeoff between the structural risk (the output weight norm) and the empirical risk (the error) by regulating a proportion of each during optimization. To achieve this tradeoff, the empirical risk in the objective function is weighted by a regulating factor.

Second, ELML2 is a regularized version of the ELM, which has all the advantages of the basic ELM[9]. Moreover, it introduces a Lagrange multiplier based constraint optimization method. Therefore, it achieves reliable performance with a different type of feature mapping.

Third, KELM is an optimized ELM, which links the ELM minimal weight norm property to the Support Vector Machine (SVM) maximal margin for classification[14]. It is shown that through standard optimization of the ELM, a so-called support vector network with a better generalization property can be obtained by the KELM. However, compared with the standard SVM, KELM has fewer optimization constraints.

Explanations of all the used base learners can be found in Refs. [9, 13, 14].

2.3 Correntropy

The generalized correlation measures the similarity between the feature vectors by studying the interaction between them[32, 33]. Let $\boldsymbol{C}' = [c'_j]_{j=1}^{T}$ and $\boldsymbol{D} = [d_j]_{j=1}^{T}$ be two arbitrary random vectors. Then, correntropy between them is

$$\boldsymbol{V}_{\sigma}(\boldsymbol{C}', \boldsymbol{D}) = E[K_{\sigma}(\boldsymbol{C}', \boldsymbol{D}) \tag{7}$$

where $K_{\sigma}(\cdot)$ represents the kernel function used in accordance with Mercer's theorem[34], and $E[\cdot]$ represents the expectation operator. The Mercer kernel function is the Gaussian kernel for all finite sequences of points $\{(c'_j, d_j)\}_{j=1}^{T}$, and the correntropy is defined by

$$\widehat{V}_{T,\sigma}(\boldsymbol{C}', \boldsymbol{D}) = \frac{1}{T}\sum_{j=1}^{T} K_{\sigma}(c'_j - d_j) \tag{8}$$

If $K_{\sigma}(\cdot)$ is given by

$$K_{\sigma}(c'_j - d_j) \triangleq G(c'_j - d_j) = \exp\left\{-\frac{(c'_j - d_j)^2}{2\sigma^2}\right\} \tag{9}$$

where σ is the kernel size, then, Eq. (8) becomes

$$\widehat{V}_{T,\sigma}(\boldsymbol{C}', \boldsymbol{D}) = \frac{1}{T}\sum_{j=1}^{T} G(c'_j - d_j) \tag{10}$$

According to Ref. [33], the Maximum Correntropy Criterion (MCC) is represented as Eq. (7). As correntropy is insensitive to outliers, it performs better than MSE in the case of disruptions within the input data[33, 35].

2.4 Negative correlation learning

NCL is a machine learning paradigm designed to enhance diversity among learners so that each

learner achieves its best performance among the ensemble[27]. Since the errors of the base learners are uncorrelated (negatively correlated) and unbiased, then, the ensemble error is

$$E_{\text{ens}} = \frac{1}{O}\sum_{h=1}^{O} E_h = \frac{1}{O}\sum_{h=1}^{O}\sum_{j=1}^{N}\left\{\frac{1}{2}\left[g_h(X_j)-Y_j\right]^2 - \left[g_h(X_j)-g_{\text{ens}}(X_j)\right]^2\right\} \tag{11}$$

where $g_h(X_j)$ is the output of the base learner, $g_{\text{ens}}(X_j)$ represents ensemble result, and O represents the number of base learners. Here, $\frac{1}{2}\left[g_h(X_j)-Y_j\right]^2$ can be regarded as the measure of MSE.

3 Proposed Ensemble (HE^{2}LM)

Ensemble learning aims to construct multiple diverse classifiers through combining their outputs. The ensemble enhances performance more than that of the base learners.

As the ELM uses random weights, it often has a low misclassification rate. Various ELM ensemble models have been proposed in Refs. [36, 37]. In this article, we employ data splitting of the training data, while a heterogeneous framework is designed, and three types of ELM algorithms are used as the base learners. Then, the ensemble is constructed by training the base classifiers on split data. A maximum weighted sum is computed to combine the output from all member classifiers into the ensemble pool. Using NCL with correntropy and different training parameters of the base ELM learning algorithms allows each member classifier to generate different decision boundaries; hence, different errors are obtained which reduce the combined error from the whole ensemble.

Since the distribution of the data is important and affects the performance of the learning classifiers, we divide the training dataset into distinct parts with the same imbalanced ratio to almost preserve the original data distribution while subsequently conducting random resampling on the dataset. Moreover, the obtained classifiers are more diverse and have different errors. For example, if we divide the training data S into 3 parts, namely $S = \{S_1, S_2, S_3\}$, we have three training subsets: $\{S_2, S_3\}$, $\{S_1, S_3\}$, and $\{S_1, S_2\}$.

A sufficient and necessary condition for the ensemble to outperform its base members is that component learners should be simultaneously accurate and diverse. Considering correntropy and NCL are both used in the HE^{2}LM to improve the performance of ensemble, the proposed model can be described as follows (Fig. 1):

- Divide the original data into parts according to the sample sizes.
- Use the random resampling technique to select the training data.
- Randomly select the base classifier from {RELM, KELM, ELML2} to train the selected data.
- Replace the MSE by correntropy in the objective function of ELM.
- Use the NCL technique in the ensemble to improve diversity among the classifiers.
- Employ the weighted sum method to test the unseen samples in the testing phase.

In summary, data division, correntropy, NCL, random resampling, and heterogeneous classifiers are used to construct the HE^{2}LM to improve performance. The description for our proposed algorithm (HE^{2}LM) is listed in Algorithm 1.

3.1 Architecture

The original training dataset is divided randomly into K subsets with equal size averagely. If we have N samples, the size of each subset will be N/K. To maximize the diversity among the reconstructed training datasets, each new training set is obtained through resampling on $(K-1)$ of K subsets. Then, each subset is trained by one base classifier selected

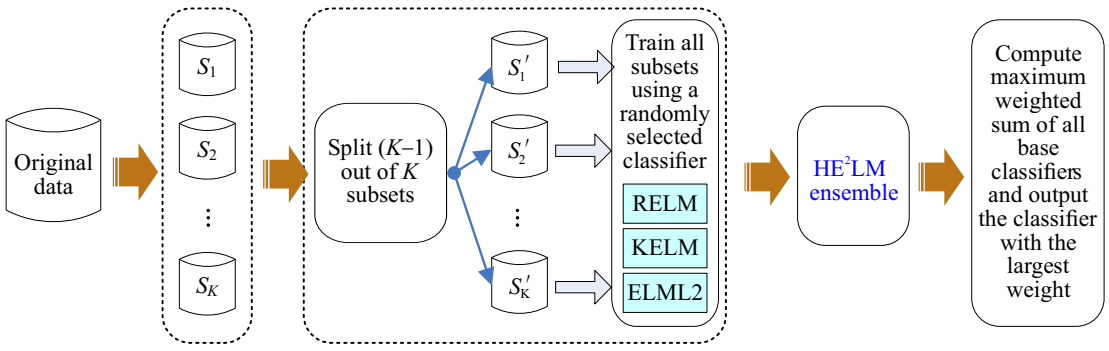

Fig. 1 The general scheme of the HE^{2}LM algorithm.

randomly out of three. The process of adding the trained classifier to the ensemble is repeated for all remaining subsets. The whole framework is shown in Fig. 1.

The MSE in the ELM function (Eq. (1)) is replaced by correntropy in Eq. (10) as it is more robust when noise exists. During the iteration, if the diversity and accuracy of the current ensemble increase, it will be retained in the updated ensemble. We replace MSE ($\frac{1}{2}\left[g_h(X_j)-Y_j\right]^2$) in Eq. (11) by correntropy ($\{G(g_h(X_j)-Y_j)\}$) to enhance the negative correlation learning between the base classifier output and the ensemble output. Then, the error of the ensemble in Eq. (11) is computed by

$$E_{\text{ens}} = \frac{1}{O}\sum_{h=1}^{O} E_h = \frac{1}{O}\sum_{h=1}^{O}\sum_{j=1}^{N}\left\{G(g_h(X_j)-Y_j) - \left[g_h(X_j)-g_{\text{ens}}(X_j)\right]^2\right\} \quad (12)$$

Then, the final ensemble model is a mixture of all classifiers trained on all subsets. After the training process is finished, the labels for the tested data are obtained by using the weighted sum method applied to the output of all member classifiers in the evolved ensemble.

Algorithm 1: HE^{2}LM

Train phase ()

Input: Original training dataset S; the number of hidden nodes L, threshold ε, the number of iterations T, and the number of subsets K.

Output: Ensemble classifier model E.

1: Split the original training dataset: $S = \{S_1, S_2, \cdots, S_K\}$;
2: **for** each i (from 1 to T) **do**
3: **for** each j (from 1 to K) **do**
4: Set $S_{\text{sub}} = S - S_j$;
5: Reconstruct training S_{tr} by resampling on S_{sub};
6: Randomly select a kind of ELM(e_j) type from the three types {RELM, KELM, ELML2};
7: Train ELM(e_j) on S_{tr};
8: Add classifier e_j to the ensemble E_K;
9: **if** (E_K_error $< \varepsilon$) **then**
10: Add it to the Ensemble E

Predict phase ()

Input: Unknown sample X, ensembles classifier model $E = \{E_1, E_2, \cdots, E_{T'}\}$;

Output: Class label of sample X.

11: Loop for $E = \{E_1, E_2, \cdots, E_{T'}\}$
12: Compute the weighted sum of all the outputs $Y = [Y_1, Y_2, \cdots, Y_{T'}]$, then output the class label of X with the highest weight.

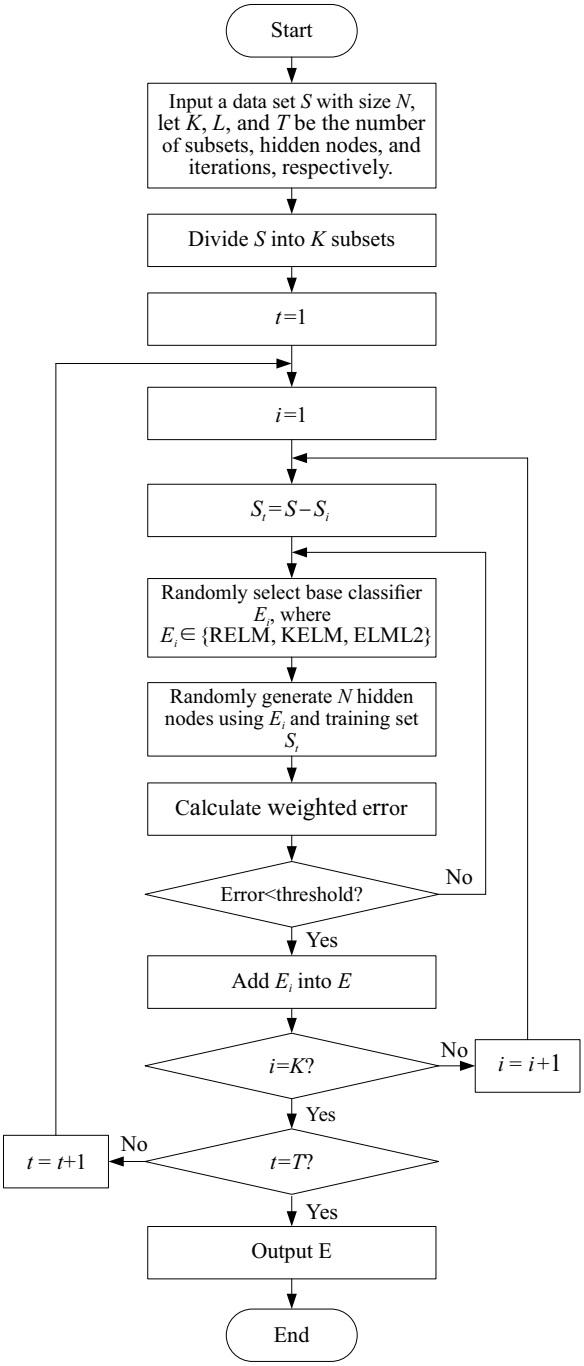

Fig. 2 Flowchart for ensemble construction in HE^{2}LM.

3.2 Implementation

The implementation for the ensemble construction and training is described in Fig. 2. Given a testing instance (y, y_t), an ensemble of T' predictors is created. For pattern y in the ensemble, we use the weighted sum to make the final decision. Suppose that there is a C-class problem, we calculate the weighted sum for all classifiers for all classes. The class that receives

the maximum weighted sum from all predictors is considered the predicted label:

$$\boldsymbol{L}(y) = \arg\max \sum_{m=1}^{T'} \alpha_m \cdot (f_m(y) = \boldsymbol{L}) \tag{13}$$

where α_m is the weight of the base learner and $f_m(y)$ is the prediction result.

4 Simulation and Discussion

4.1 Simulation settings

To test the performance of the HE^{2}LM, we conduct the simulations on datasets from a machine learning repository (UCI)[38]. The simulations are for both classification and regression datasets. Tables 1 and 2 show the descriptions for the classification and regression datasets, respectively. More details of the datasets can be found on the web pages of those repositories. Simulations are conducted in MATLAB 8.1.0, using an Intel Core i5 processor with 2.4 GHz CPU and 4 GB RAM. To remove any biases from the results, we repeat the simulation and compute the average accuracy for all iterations. Using random resampling, the training data is split into 2–8 subsets with equal size, according to the number of instances in the datasets. The error function is the error rate of all misclassified samples.

Table 1 Descriptions for the classification datasets.

Dataset	Number of features	Number of train	Number of test	Number of classes
Balance	5	312	313	3
Dermatology	35	286	72	6
USPS	256	7792	1506	10
Isolet	618	6238	1559	26
Hayas	5	132	28	3
Climate	18	390	150	2
Hepatitis	19	120	30	2
Pima	8	615	153	2
Liver	6	245	100	2
Vowel	13	781	209	10
Credit	24	796	204	2

Table 2 Descriptions for the regression datasets.

Dataset	Number of features	Number of train	Number of test
Servo	5	83	83
Yacht	6	154	154
Stock	8	267	267
Self-Noise	6	752	751
Slump	10	52	51

To evaluate our method, we use average accuracy to measure the generalization performance as an indication of the classification output correctness. The cost of training new test data should not have a significant effect on the ensemble accuracy when we train the ensemble with any training set, whose size is a bit more, or less than the original data. Standard deviation of the accuracy rates is used as an indication of the ensemble's stability, where lower standard deviation indicates a more stable method.

4.2 Synthetic data

We test the proposed method on two synthetic datasets for classification and regression as follows.

The Two-Moon synthetic dataset for classification contains 200 data samples. The 100 positive samples are generated by

$$f(n) = \begin{cases} X_1 = \cos(z), \\ X_2 = \sin(z), \end{cases}$$

and the negative samples are generated by

$$f(n) = \begin{cases} X_1 = 1 + \cos(z), \\ X_2 = \frac{1}{2} - \sin(z), \end{cases}$$

where $z \in (0, \pi)$.

To test the sensitivity to noise, we randomly chose different percentages of the training data in each dataset and disrupted their target labels randomly by converting the class label sign, i.e., 1 to -1, or -1 to 1. Figure 3 shows the distribution of the classes.

For the Sinc synthetic dataset for regression which

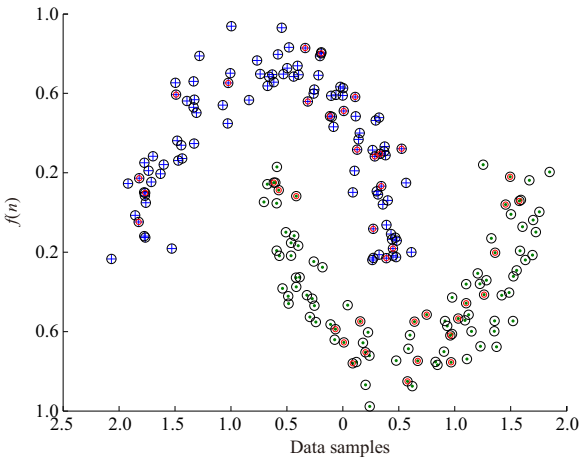

Fig. 3 The distribution of the positive and negative classes in the Two-Moon dataset, where $\oplus$ represents the positive data, $\odot$ represents the negative data, and red nodes represent the noise.

contains 200 data samples, the data samples were generated using the function $f(x) = \frac{\sin(x)}{x} + \upsilon$, where υ represents the Gaussian noise. The samples are drawn uniformly from $[-5, 5]$ for each noise level.

4.3 User-specified parameters

The classification and regression simulations on the datasets are performed using the ELM-RCC[28], RELM[13], and HE^{2}LM algorithms. To attain better generalization, the parameters of the base ELM classifiers and regressors (RELM, KELM, and ELML2), C, kernel parameter λ, and σ for Gaussian should be carefully selected. During the simulations, we test different values of C, λ, and σ upon all datasets. The range of λ is $\{0.1, 0.2, \cdots, 10\}$, the range of C is $\{0.2, 0.4, 0.5, 2, 4, \cdots, 50\}$, and the range of σ is $\{1.1, 1.2, \cdots, 14\}$. The number of hidden nodes is selected from the range $\{10, 20, \cdots, 1000\}$. Tables 3 and 4 show the optimum case of the selected parameters for classification and regression, respectively.

Table 3 Parameters used by ELM-RCC, RELM, and HE^{2}LM in the classification datasets. Here, "nh" means the number of hidden nodes.

Dataset	ELM-RCC (C, σ, nh)	RELM (C, nh)	HE^{2}LM (C, σ, nh, λ)
Balance	(10, 9, 450)	(0.2, 15)	(6, 7, 600, 0.1)
Dermatology	(6, 7, 350)	(0.2, 30)	(6, 7, 600, 1.8)
USPS	(30, 9, 600)	(50, 1000)	(30, 7, 1000, 0.3)
Isolet	(10, 9, 450)	(30, 700)	(30, 7, 1000, 0.3)
Hayes	(20, 8, 500)	(0.5, 50)	(20, 5, 330, 0.9)
Climate	(30, 14, 600)	(20, 300)	(36, 8, 300, 9)
Hepatitis	(8, 5, 100)	(4, 100)	(6, 9, 300, 1.1)
Pima	(4, 3, 30)	(16, 20)	(8, 8, 20, 0.9)
Liver	(22, 4, 30)	(14, 70)	(26, 6, 40, 0.9)
Vowel	(6, 3, 60)	(8, 40)	(12, 3, 30, 0.6)
Credit	(28, 0.4, 150)	(12, 120)	(16, 4, 170, 1.3)

Table 4 Parameters used by ELM-RCC, RELM, and HE^{2}LM in the regression datasets. Here, "nh" means the number of hidden nodes.

Dataset	ELM-RCC (C, σ, nh)	RELM (C, nh)	HE^{2}LM (C, σ, nh, λ)
Servo	(0.2, 0.9, 50)	(4, 30)	(10, 3, 20, 0.6)
Yacht	(2, 2, 40)	(6, 30)	(12, 6, 70, 1.2)
Stock	(4, 6, 90)	(18, 110)	(22, 6, 100, 0.8)
Self-noise	(14, 2, 90)	(16, 40)	(14, 7, 80, 0.6)
Slump	(4, 0.2, 20)	(22, 140)	(20, 9, 220, 0.3)

4.4 Simulation results

The average error rates of the simulations for classification and regression are shown in Tables 5 and 6. These tables show that our method achieves the lowest error rates in almost all cases compared with the ELM-RCC, and in most cases compared with the RELM. Compared with the ELM-RCC and the RELM, the relative error reduction is 18% and 17%, respectively, upon the classification datasets, and it is 7% and 23%, respectively, upon the regression datasets. Meanwhile, we also provide a comparison for the standard deviation of the accuracy rates in Tables 7 and 8 for the classification and regression datasets, respectively. We observe that the standard deviation of the accuracy rates of the HE^{2}LM is better than those of the RELM in almost all datasets and approximately the same as those in the ELM-RCC.

For the Two-Moon synthetic dataset, the testing results of the ELM-RCC, RELM, and HE^{2}LM algorithms are shown in Fig. 4–6, respectively. These figures indicate that the HE^{2}LM algorithm produces a smoother boundary compared with both the ELM-RCC and RELM algorithms.

Table 5 Error rates of ELM-RCC, RELM, and HE^{2}LM upon the classification datasets.

Dataset	ELM-RCC	RELM	HE^{2}LM
Balance	0.0900	0.0895	**0.0777**
Dermatology	0.0500	0.0500	**0.0417**
USPS	0.0691	0.0400	**0.0598**
Isolet	0.0733	0.0990	**0.0475**
Hayes	0.2600	0.1500	**0.2143**
Climate	0.1044	0.1089	**0.0956**
Hepatitis	0.1420	0.1400	**0.1200**
Pima	0.3540	0.3652	**0.3185**
Liver	0.3000	0.3533	**0.2500**
Vowel	0.3400	0.3500	**0.3200**
Credit	0.2352	0.2418	**0.2156**

Table 6 Error rates of the ELM-RCC, RELM, and HE^{2}LM algorithms upon the regression datasets.

Dataset	ELM-RCC	RELM	HE^{2}LM
Servo	0.7700	0.7850	**0.6375**
Yacht	9.5457	12.8740	**9.1094**
Stock	0.0045	0.0046	**0.0045**
Self-Noise	6.4511	8.7476	**6.1677**
Slump	3.2800	3.4100	**3.1762**

Table 7 Standard deviation of accuracy rates of the ELM-RCC, RELM, and HE^{2}LM algorithms in the classification datasets.

Dataset	ELM-RCC	RELM	HE^{2}LM
Balance	0.0067	0.0166	**0.0121**
Dermatology	0.0158	0.0212	**0.0139**
USPS	0.0848	0.0171	**0.0063**
Isolet	0.0055	0.0287	**0.0006**
Hayes	0.0160	0.1288	**0.1351**
Climate	0.0038	0.0102	**0.0038**
Hepatitis	0.0552	0.0588	**0.0451**
Pima	0.0350	0.3310	**0.0299**
Liver	0.0510	0.0503	**0.0469**
Vowel	0.0386	0.0522	**0.0208**
Credit	0.0186	0.0143	**0.0264**

Table 8 Standard deviation of accuracy rates of ELM-RCC, RELM, and HE^{2}LM algorithms in the regression datasets.

Dataset	ELM-RCC	RELM	HE^{2}LM
Servo	0.1185	0.1820	**0.1298**
Yacht	0.1073	0.1089	**0.1114**
Stock	0.000 04	0.000 06	**0.000 01**
Self-Noise	0.1300	0.1305	**0.1091**
Slump	0.9225	0.8801	**0.7224**

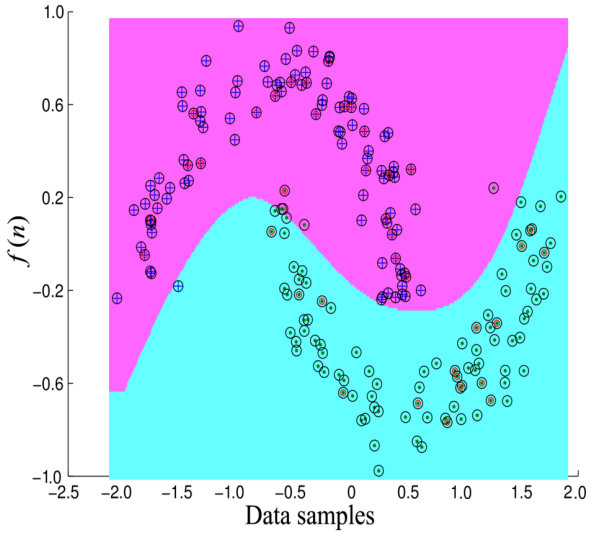

Fig. 4 The classification results of RELM algorithm upon the Two-Moon dataset.

For the Sinc synthetic dataset, the testing results of the ELM-RCC, RELM, and HE^{2}LM algorithms with noise, $\upsilon \sim N(0, 0.1)$ are shown in Fig. 7. From Fig. 7, the HE^{2}LM algorithm is more robust against noise compared with both the ELM-RCC and RELM algorithms as the regression errors of HE^{2}LM, ELM-RCC, and RELM algorithms are 0.0985, 0.1082, and 0.1020, respectively.

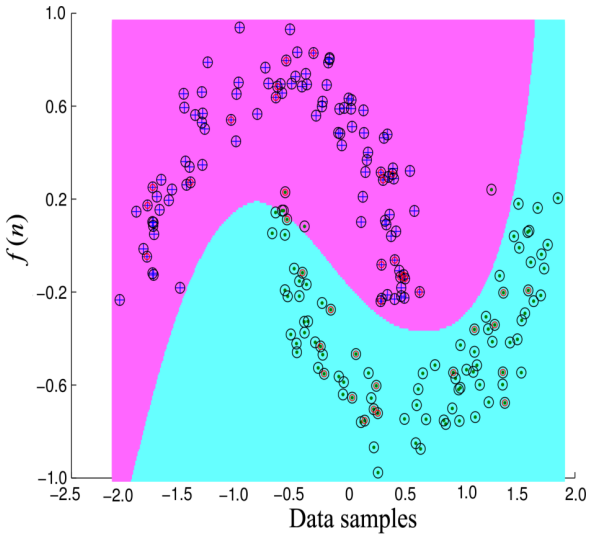

Fig. 5 The classification results of ELM-RCC algorithm upon the Two-Moon dataset.

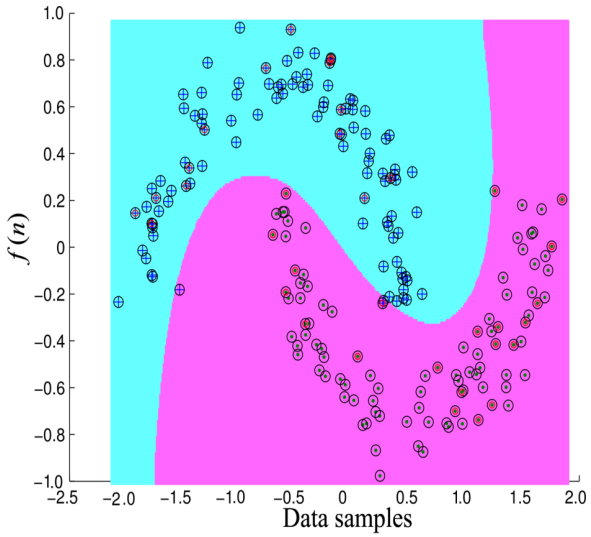

Fig. 6 The classification results of HE^{2}LM algorithm upon the Two-Moon dataset.

Figures 8 and 9 demonstrate that the HE^{2}LM algorithm outperforms both the ELM-RCC and RELM algorithms.

5 Conclusion

In this article, we propose an advanced approach for classification using a heterogeneous ensemble, namely the HE^{2}LM ensemble to deal with noisy data. To achieve diversity within the proposed HE^{2}LM ensemble, different ELM algorithms, i.e., RELM, KELM, and ELML2, are integrated and each one is

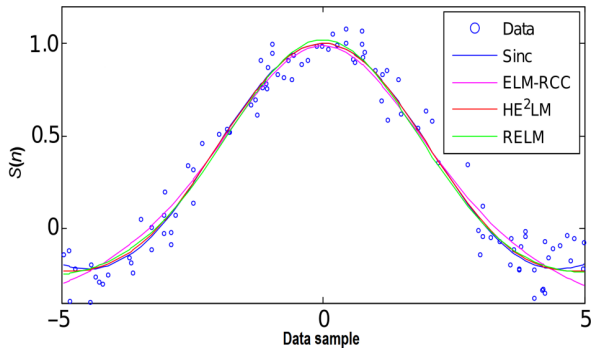

Fig. 7 Display of the regression results of HE^{2}LM, ELM-RCC, and RELM algorithms upon the Sinc dataset with Gaussian noise $\upsilon \sim N(0, 0.1)$ ($S(n) = \frac{\sin(n)}{n} + \upsilon$).

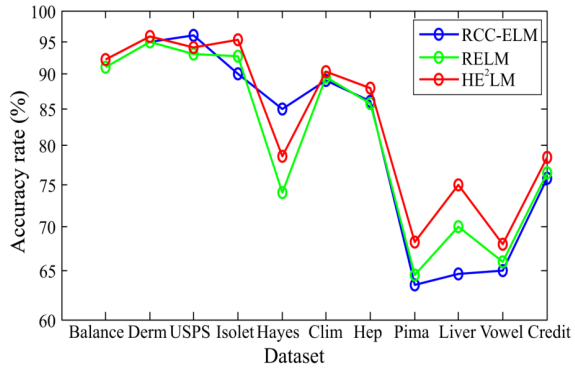

Fig. 8 Display of accuracy rates results of HE^{2}LM, ELM-RCC, and RELM algorithms upon the classification dataset. Here, "Derm", "Clim", and "Hep" mean Dermatology, Climate, and Hepatitis datasets.

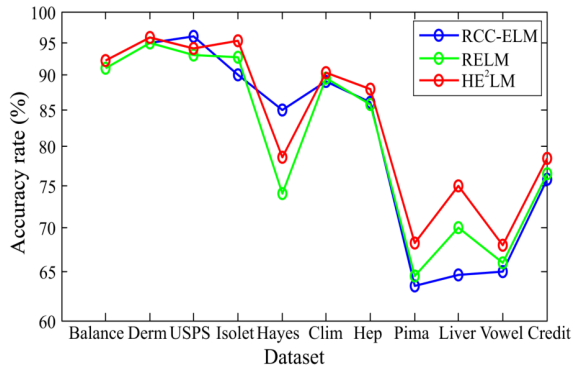

Fig. 9 Display of error rates results of HE^{2}LM, ELM-RCC, and RELM algorithms upon the regression dataset.

independent of the other. To enhance the accuracy rate in the proposed ensemble, we learned various parts of the original training dataset with different types of ELM classifiers. In the proposed HE^{2}LM algorithm, we replaced MSE by correntropy in the ELM objective function, and employed a negative correlation in the learning process to produce a more robust ensemble against noise. Moreover, we employed a random resampling technique in the training data to allow the base classifiers to generate different decision boundaries and different errors, while reducing the total error. Hence, the final ensemble is less sensitive to noise and achieves better generalization performance. Simulation results on benchmark and synthetic datasets result in higher accuracy rates and lower standard deviations compared with the ELM-RCC and RELM algorithms and verify the effectiveness of the proposed HE^{2}LM ensemble.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (Nos. 61174103 and 61603032), the National Key Technologies R&D Program of China (No. 2015BAK38B01), the National Key Research and Development Program of China (No. 2017YFB0702300), and the China Postdoctoral Science Foundation (No. 2016M590048), and the University of Science and Technology Beijing – Taipei University of Technology Joint Research Program (TW201705).

References

[1] T. Löfström, On effectively creating ensembles of classifiers: Studies on creation strategies, diversity and predicting with confidence, PhD dissertation, Dept. Comput. Syst. Sci., Stockholm University, Swedish, 2015.

[2] N. Perez-Diaz, D. Ruano-Ordas, F. Fdez-Riverola, and J. R. Mendez, Boosting accuracy of classical machine learning antispam classifiers in real scenarios by applying rough set theory, *Sci. Program.*, doi: 10.1155/2016/5945192.

[3] L. Breiman, *Bagging Predictors*. Boston, MA, USA: Kluwer Academic Publishers, 1996.

[4] Y. Freund, R. Schapire, and N. Abe, A short introduction to Boosting, *J. Japanese Soc. Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1999.

[5] Y. Freund and R. E. Schapire, Experiments with a new Boosting algorithm, in *Proc. 13th Int. Conf. Machine Learning*, Bari, Italy, 1996, pp. 1–9.

[6] X. Luo, J. Liu, D. D. Zhang, and X. H. Chang, A large-scale web QoS prediction scheme for the industrial Internet of Things based on a kernel machine learning algorithm, *Comput. Networks*, vol. 101, pp. 81–89, 2016.

[7] X. Luo, J. Deng, J. Liu, W. P. Wang, X. J. Ban, and J. H. Wang, A quantized kernel least mean square scheme with entropy-guided learning for intelligent data analysis, *China Commun.*, vol. 14, no. 7, pp. 127–136, 2017.

[8] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, OP-ELM: Optimally pruned extreme learning machine, *IEEE Trans. Neural Networks*, vol. 21, no. 1, pp. 158–162, 2010.

[9] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, Extreme learning machine for regression and multiclass classification, *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 42, no. 2, pp. 513–529, 2012.

[10] M. Atiquzzaman and J. Kandasamy, Prediction of hydrological time-series using extreme learning machine, *J. Hydroinform.*, vol. 18, no. 2, pp. 345–353, 2016.

[11] Y. Xu, X. Luo, W. P. Wang, and W. B. Zhao, Efficient DV-HOP localization for wireless cyber-physical social sensing system: A correntropy-based neural network learning scheme, *Sensors*, vol. 17, no. 1, p. 135, 2017.

[12] X. Luo, D. D. Zhang, L. T. Yang, J. Liu, X. H. Chang, and H. S. Ning, A kernel machine-based secure data sensing and fusion scheme in wireless sensor networks for the cyber-physical systems, *Future Gener. Comput. Syst.*, vol. 61, pp. 85–96, 2016.

[13] W. Deng, Q. Zheng, and L. Chen, Regularized extreme learning machine, in *Proc. IEEE Symp. Comput. Intell. Data Min.*, Nashville, TN, USA, 2009, pp. 389–395.

[14] G. B. Huang, X. Ding, and H. Zhou, Optimization method based extreme learning machine for classification, *Neurocomputing*, vol. 74, nos. 1–3, pp. 155–163, 2010.

[15] X. Luo and X. H. Chang, A novel data fusion scheme using grey model and extreme learning machine in wireless sensor networks, *Int. J. Control Autom. Syst.*, vol. 13, no. 3, pp. 539–546, 2015.

[16] X. Luo, X. H. Chang, and X. J. Ban, Regression and classification using extreme learning machine based on L1-norm and L2-norm, *Neurocomputing*, vol. 174, pp. 179–186, 2016.

[17] H. Yu, Y. Yuan, X. Yang, and Y. Dan, A dynamic generation approach for ensemble of extreme learning machines, *Lect. Notes Comput. Sci.*, vol. 8866, pp. 294–302, 2014.

[18] N. Liu and H. Wang, Ensemble based extreme learning machine, *IEEE Signal Process. Lett.*, vol. 17, no. 8, pp. 754–757, 2010.

[19] A. Samat, P. Du, S. Liu, J. Li, and L. Cheng, E^2LMs: Ensemble extreme learning machines for hyperspectral image classification, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 7, no. 4, pp. 1060–1069, 2014.

[20] H. Wang, Q. He, T. Shang, F. Zhuang, and Z. Shi, Extreme learning machine ensemble classifier for large-scale data, in *Proc. ELM*, Singapore, 2014, pp. 151–161.

[21] S. Huang, B. Wang, J. Qiu, J. Yao, G. Wang, and G. Yu, Parallel ensemble of online sequential extreme learning machine based on MapReduce, *Neurocomputing*, vol. 174, pp. 352–367, 2014.

[22] J. Cao, T. Chen, and J. Fan, Landmark recognition with compact BoW histogramand ensemble ELM, *Multimed. Tools Appl.*, vol. 75, no. 5, pp. 2839–2857, 2016.

[23] Y. Jin, J. Cao, Y. Wang, and R. Zhi, Ensemble based extreme learning machine for cross-modality face matching, *Multimed. Tools Appl.*, vol. 75, no. 19, pp. 1–16, 2016.

[24] X. L. Wang, Y. Y. Chen, H. Zhao, and B. L. Lu, Parallelized extreme learning machine ensemble based on min max modular network, *Neurocomputing*, vol. 128, no. 5, pp. 31–41, 2014.

[25] A. O. M. Abuassba, D. Zhang, X. Luo, A. Shaheryar, and H. Ali, Hazrat, Improving classification performance through an advanced ensemble based heterogeneous extreme learning machines, *Comput. Intell. Neurosci.*, doi: 10.1155/2017/3405463.

[26] H. Lu, J. W. Zhang, X. Ma, and W. Zheng, Tumor classification using extreme learning machine ensemble, (in Chinese), *Math. Pract. Theory*, vol. 42, no. 17, pp. 148–154, 2012.

[27] Y. Liu and X. Yao, Ensemble learning via negative correlation, *Neural Netw.*, vol. 12, no. 10, pp. 1399–1404, 1999.

[28] H. J. Xing and X. M. Wang, Training extreme learning machine via regularized correntropy criterion, *Neural Comput. Appl.*, vol. 23, nos. 7&8, pp. 1977–1986, 2013.

[29] K. Li, X. Kong, Z. Lu, W. Liu, and J. Yin, Boosting weighted ELM for imbalanced learning, *Neurocomputing*, vol. 128, pp. 15–21, 2014.

[30] Y. Jiang, Y. Shen, Y. Liu, and W. Liu, Multiclass Adaboost ELM and its application in LBP based face recognition, *Math. Probl. Eng.*, doi: 10.1155/2015/918105.

[31] Y. Zhang, J. Wu, Z. Cai, P. Zhang, and L. Chen, Memetic extreme, learning machine, *Pattern Recogn.*, vol. 58, pp. 135–148, 2016.

[32] I. Santamaia, P. P. Pokharel, and J. C. Principe, Generalized correlation function: Definition, properties and application to blind equalization, *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2187–2197, 2006.

[33] W. Liu, P. P. Pokharel, and J. C. Principe, Correntropy: Properties and applications in non-Gaussian signal processing, *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5286–5298, 2007.

[34] V. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer, 1995.

[35] X. Luo, J. Deng, W. P. Wang, J. H. Wang, and W. B. Zhao, A quantized kernel learning algorithm using a minimum kernel risk-sensitive loss criterion and bilateral gradient technique, *Entropy*, vol. 19, no. 7, p. 365, 2017.

[36] G. Wang and P. Li, Dynamical Adaboost ensemble extreme learning machine, in *Proc. 3rd Int. Conf. Adv. Comput. Theory Eng.*, Chengdu, China, 2010, pp. V354–V358.

[37] V. Sachnev, S. Ramasamy, S. Sundaram, H. J. Kim, and H. J. Hwang, A cognitive ensemble of extreme learning machines for steganalysis based on risk-sensitive hinge loss function, *Cognitive Comput.*, vol. 7, no. 1, pp. 103–110, 2014.

[38] D. Newman, S. Hettich, C. Blake, C. Merz, and D. Aha, UCI repository of machine learning databases, http://archive.ics.uci.edu/ml/datasets.html, 2017, Jul. 30.

Adnan O. M. Abuassba is currently working toward his PhD degree in computer science at the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. He received the master degree of computer science (MSc) from Al-Quds University, Palestine in 2007. His current research interests include neural networks, machine learning, ensemble learning, and computational intelligence.

Xiong Luo received the PhD degree from Central South University, China, in 2004. From 2004 to 2005, he received post doctoral fellowships from Tsinghua University, China. From 2012 to 2013, he was at Arizona State University, USA, as a visiting scholar. He currently works as a professor in the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research interests include machine learning, cloud computing, and computational intelligence. He has published extensively in his areas of interest in journals, such as the Future Generation Computer Systems, Neurocomputing, Computer Networks, Cognitive Computation, and Personal and Ubiquitous Computing.

Dezheng Zhang received the PhD degree from University of Science and Technology Beijing, China in 2002, where he works as a professor. His current research interests include knowledge engineering and data mining. He has published extensively in his areas of interest.

Wulamu Aziguli received the PhD degree from University of Science and Technology Beijing, China in 2004, where she works as a professor. Her current research interests include data mining and machine learning.

Yao Zhang is currently working toward her master degree at New York University, USA. She received the bachelor degree from the University of California, Santa Barbara, USA in 2017. Her research interest includes statistical arbitrage and big data analytics.