Conventional kernel support vector machine (KSVM) has the problem of slow training speed, and single kernel extreme learning machine (KELM) also has some performance limitations, for which this paper proposes a new combined KELM model that build by the polynomial kernel and reproducing kernel on Sobolev Hilbert space. This model combines the advantages of global and local kernel function and has fast training speed. At the same time, an efficient optimization algorithm called cuckoo search algorithm is adopted to avoid blindness and inaccuracy in parameter selection. Experiments were performed on bi-spiral benchmark dataset, Banana dataset, as well as a number of classification and regression datasets from the UCI benchmark repository illustrate the feasibility of the proposed model. It achieves the better robustness and generalization performance when compared to other conventional KELM and KSVM, which demonstrates its effectiveness and usefulness.

Keywords: kernel extreme learning machine, combined kernel function, reproducing kernel function, cuckoo search algorithm

1. Introduction

Based on the conditions of support vector machine (SVM) learning principle, G. B. Huang proposed kernel extreme learning machine in 2010 [1], which introduced the kernel functions to extreme learning machine (ELM). It provide more stable and better generalization performance than SVM and basic ELM [2]. KELM has been widely applied in the problems of classification, regression [3] and practical applications [4–9].

Learning capability and generalization performance of ELM are mainly determined by the type of kernel function and the kernel parameters. In [10], gaussian KELM model with fast local learning capability is selected. Reference [11] further optimizes the gaussian by active operators particle swam optimization algorithm. But using a single gaussian kernel may not be sufficient to solve a complex problem satisfactorily because of its global generalization performance is weak and easy to fall into local optimum. In [12], a combined KSVM model is proposed, which combines the advantages of global kernel and local kernel. The learning capability and generalization performance of this model are better than a single kernel. However, it still has the problem of low training efficiency and slow learning speed, and parameters of this model are difficult to choose effectively. To sum up, there is a need for a model that has both good performance, fast speed and efficient parameter optimization algorithm.

In this paper, a new combined kernel function model called Rep_P KELM (Reproducing Polynomial kernel extreme learning machine) is proposed which is built by the reproducing kernel on Sobolev Hilbert [13] and the polynomial kernel. Reproducing kernel has played an important role in the theory of function approximation and regularization [14], however, it is still in-depth to apply reproducing kernel to the problems of classification and regression of KELM. It is first strictly proved that reproducing kernel can be used as an allowed extreme learning machine kernel function in theory. Then the model applied to the simulation experiments show that the proposed model has better robustness and generalization performance in classification and regression of KELM when compared with other conventional KELM and KSVM. This provides a new method for further study of KELM and extends the field of KELM theory.

2. Kernel Extreme Learning Machine

Given a training set: \( \mathbb{R} = \{(x_i, y_i) | x_i \in \mathbb{R}^d, y_i \in \mathbb{R}^m, i = 1, \ldots, N\} \), the vector of the output weights \( b = [b_1, \ldots, b_L]^T \), hidden node output function \( h(x) = [h_1(x), \ldots, h_L(x)] \) and hidden node number \( L \). The mathematical model of ELM can be generated as follows:

\[
    f(x) = \sum_{i=1}^{L} b_i h_i(x) = h(x)b. \quad \ldots \quad \ldots \quad \ldots \quad (1)
\]
The optimal output weight can be analytically determined as [15, 16]:

\[ b = H^T \left( \frac{I}{C} + HH^T \right)^{-1} T. \ldots \ldots \ldots \ldots \quad (2) \]

Where C is the regularization coefficient, T is the target vector and H is the hidden layer output matrix:

\[ H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_n) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \ldots & h_L(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \ldots & h_L(x_N) \end{bmatrix}. \ldots \ldots \ldots \ldots \quad (3) \]

The output function of KELM is established:

\[ f(x) = h(x)H^T \left( \frac{I}{C} + HH^T \right)^{-1} T. \ldots \ldots \ldots \ldots \quad (4) \]

If the matrix H in hidden layer is unknown, kernel matrix can be defined as follow at the condition of Mercer [17]:

\[ \Omega_{ELM} = HH^T : \Omega_{ELM,i,j} = h(x_i) \cdot h(x_j) = K(x_i, x_j). \quad (5) \]

Finally, the output function of KELM can be represented as:

\[ f(x) = \begin{bmatrix} K(x,x_1) \\ \vdots \\ K(x,x_n) \end{bmatrix} \left( \frac{I}{C} + \Omega_{ELM} \right)^{-1} T. \ldots \ldots \ldots \ldots \quad (6) \]

Common properties and theorems of KELM are given below:

**Theorem 1.** (Mercer Theorem) [18]. Assume that for \( X \subset \mathbb{R}^n \), \( K(x, z) \) is a continuous symmetric real value function on \( X \times X \) such that the following integration should always be non-negative for every \( f \in L_2(X) \):

\[ \int_{X \times X} K(x,z) f(x) f(z) dx dz \geq 0. \]

Then \( K(x, z) \) must be a valid kernel function.

**Theorem 2.** [19]. The necessary and sufficient condition for the translation invariant function \( K(x, z) = f(x - z) \) as a kernel function is: \( f(0) > 0 \), where \( f: X \rightarrow \mathbb{R} \). And Fourier transform is \( F[K](\omega) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} \exp(-i\omega x) K(x) dx \geq 0. \)

**Property 1.** [19]. Assume that \( K_1(x, z) \) and \( K_2(x, z) \) are valid kernel on \( X \times X \), then kernel \( K(x, z) = K_1(x, z) + K_2(x, z) \) and \( K(x, z) = K_1(x, z) \times K_2(x, z) \) also are valid on \( X \times X \).

It is an allowed KELM when satisfies the Theorem 1 (Mercer Theorem), which is the same as SVM [3]. The kernel function can be a translation invariant form when it satisfies the Theorem 2, we can say it also an allowed KELM.

3. Reproducing KELM on the Sobolev Hilbert Space \( H^1(\mathbb{R}; a, b) \)

**Definition 1.** [20]. On the Sobolev Hilbert space \( H^1(\mathbb{R}; a, b) \) on \(( -\infty, +\infty) \) comprising all comply valued and absolutely continuous functions \( f(x) \) with finite norms \( \int_{-\infty}^{+\infty} (a^2 |f'(x)|^2 + b^2 |f(x)|^2) dx \)^{1/2} < \infty \), where \( a, b > 0, f'(x) \) is the derivative of \( f(x) \). The function

\[ G_{a,b}(x,z) = \frac{1}{2ab} e^{-\frac{b}{2}|x-z|} = \frac{1}{2\pi} \int_{\mathbb{R}} e^{i\omega(x-z)} a^2 \omega^2 + b^2 d\omega, \quad (7) \]

is the reproducing kernel of \( H^1(\mathbb{R}; a, b) \).

On the Hilbert space, the translation invariant kernel function form is established:

\[ K(x,z) = K(x - z) = \prod_{i=1}^{d} G_{a,b}(x_i - z_i). \ldots \ldots \ldots \ldots \quad (8) \]

Equation (8) is strictly proved to be a valid KELM kernel function in the following proof.

**Property 2.** The function \( K(x,z) = \prod_{i=1}^{d} G_{a,b}(x_i, z_i) = \prod_{i=1}^{d} G_{a,b}(x_i - z_i) \) is a valid extreme learning machine kernel function.

**Proof:**

\[ F[K](\omega) = (2\pi)^{-\frac{n}{2}} \int_{\mathbb{R}^n} \exp(-j\omega x) K(x) dx \]

\[ = (2\pi)^{-\frac{n}{2}} \int_{\mathbb{R}^n} \exp(-j\omega x) \prod_{i=1}^{n} \left( \frac{1}{2ab} e^{-\frac{b}{2}|x_i|} \right) dx \]

\[ = (2\pi)^{-\frac{n}{2}} \prod_{i=1}^{n} \left( \frac{1}{2ab} \int_{-\infty}^{+\infty} e^{-\frac{b}{2}|x_i| - j\omega x_i} dx_i \right), \quad (9) \]

\[ \frac{1}{2ab} \int_{-\infty}^{+\infty} e^{-\frac{b}{2}|x| - j\omega x} dx \]

\[ = \frac{1}{2ab} \left[ \int_{0}^{+\infty} e^{-(\frac{b}{a} + j\omega)x} dx + \int_{-\infty}^{0} e^{-(\frac{b}{a} - j\omega)x} dx \right] \]

\[ = \frac{1}{2ab} \left[ \left. \frac{1}{\frac{b}{a} + j\omega} e^{-(\frac{b}{a} + j\omega)x} \right|_{0}^{+\infty} + \left. \frac{1}{\frac{b}{a} - j\omega} e^{-(\frac{b}{a} - j\omega)x} \right|_{-\infty}^{0} \right] \]

\[ = \frac{1}{2ab} \left( \frac{1}{\frac{b}{a} + j\omega} + \frac{1}{\frac{b}{a} - j\omega} \right) \]

\[ = \frac{1}{2ab} \left( \frac{2 \cdot \frac{b}{a}}{\left(\frac{b}{a}\right)^2 + \omega^2} \right) \]

\[ = \frac{1}{\frac{b^2}{a^2} + \omega^2} \geq 0. \quad (10) \]
4. Combined Kernel Function Theory Based on Reproducing Kernel

Local and global kernel are two kinds of typical kernel functions [17]. The polynomial kernel can be presented as follow, where $d$ is kernel parameter:

$$ K_{poly}(x, z) = (x \cdot z + 1)^d . \ldots \ldots \ldots . \ldots (13) $$

Figure 2 shows that polynomial kernel belongs to global kernel that generalization performance is strong and learning ability is weak. According to Property 1, the combined kernel function is shown as follows, where

$$ m \in [0, 1] \text{ is the combined weight coefficient.} $$

$$ K_{mix} = m \cdot K_{Global} + (1 - m) \cdot K_{Local} . \ldots \ldots (14) $$

Many researches focus on the combined kernel function of gaussian and polynomial kernel [21, 22]. In this paper, reproducing kernel is used instead of gaussian kernel to combine polynomial kernel because its stronger local learning capability from Fig. 1. Thus, the model based on combined kernel function is

$$ f(x) = \begin{bmatrix} K_{mix}(x, x_1) \\ \vdots \\ K_{mix}(x, x_N) \end{bmatrix}^T \begin{bmatrix} I + K_{mix}(x_i, x_j) \end{bmatrix}^{-1} T. (15) $$

In order to reduce the complexity of the parameters, this paper take $a \times b = 1$ and removed $a$. Obviously, the training parameters of this model need to be optimized together is $(C, b, d, m)$, where $C$ is regularization coefficient, $b$ and $d$ is combined kernel parameters, and $m$ is combined weight coefficient. Many researches, include reference [12], assign these parameters through their own experience. Here, we use cuckoo search algorithm to make sure these parameters more efficient and accurate.

5. Optimization of Combined Kernel Function Parameters Based on Cuckoo Search Algorithm

Cuckoo search (CS) algorithm [23] is inspired by the behavior of parasitic cuckoo that laying their eggs in nests of other host birds. In CS algorithm, $x_i^{(t)}$ represents the position of the $i$-th nest in $t$-th generation, $x_i^{(t+1)}$ represents the new nest position, the nest position of is updated as:

$$ x_i^{(t+1)} = x_i^{(t)} + \alpha \odot L(\lambda), \quad i = 1, 2, \ldots, n . \ldots (16) $$

Where $\alpha$ is the step size, $\odot$ represents entry-wise multiplication, $L(\lambda)$ means represents the random search path.
Fig. 3. Boundaries of Rep\_P KELM on spiral.

Suppose the host bird discovers the cuckoo egg with probability is $P_a$, $P_a \in [0, 1]$. Here we refer to the method of reference [24], the specific steps of using CS algorithm to optimize the model parameters $(C, b, d, m)$ are as follows:

Step 1: Initialize the CS algorithm and set number of nests, $N$, the maximum iterations, $t_{max}$, the probability parameters, $P_a$, and the ranges of $(C, b, d, m)$.

Step 2: Generate nest positions randomly, and each nest position corresponds to a set of parameters $(C, b, d, m)$.

Step 3: Evaluate the fitness value of nest positions, find the current best solution and its corresponding position.

Step 4: Update the other nest positions using Eq. (16), and evaluate the fitness value of new positions. Compare it with the previous generation, replace the worse solution.

Step 5: Compare a random number $r$ with $P_a$, $r$ is the probability of egg detection. If $r > P_a$, randomly change the position of the nest to obtain a new group of positions.

Step 6: Find the best nest position corresponds to a set of parameters $(C, b, d, m)$ in Step 6 and output it. Stop searching when maximum iteration number limit is reached, otherwise, return to Step 3.

### 6. Experiments and Analysis

Three simulation experiments are carried out to evaluate the model. It is first proved on a challenging classification dataset, Bi-spiral dataset, that Rep\_P KELM can be used for classification. Followed by a comparison of the classification accuracy between Rep\_P KELM and single KELM on Banana dataset. At last, the performance of Rep\_P KELM and conventional model is further compared on the UCI benchmark repository in two aspects: classification and regression, the conventional model include Gaussian kernel (Gauss KELM), Gaussian Polynomial kernel (Gau\_P KELM) extreme learning machine and Reproducing Polynomial kernel support vector machine (Rep\_P KSVM) of reference [12].

#### 6.1. Classification on Bi-Spiral Dataset

Bi-spiral problem has a good ability to detect the performance of the classification algorithm. Bi-spiral can be generated by the following formula

\[
\begin{align*}
    x &= (k_1 \theta + e_1) \cos \theta \\
    y &= (k_2 \theta + e_2) \sin \theta.
\end{align*}
\]

Where $k_1 = k_2 = 0.1 \ast (\frac{\pi}{10} i)$, $e_1 = e_2 = 0.02$, $\theta \in (0, 4\pi)$. 400 training samples and 200 testing samples were selected in the noisy and no noise experiment. Figure 3 shows that Rep\_P KELM can be used in the bi-spiral problem. The classification accuracy is shown in Table 1, on spiral dataset with no noise, the training and testing accuracy can both achieve 100%, and when dataset with noise, the training accuracy is 100% and the testing accuracy is 96%.

### 6.2. Classification Performance Comparison on Banana Dataset

Banana dataset was selected to compare the classification ability of combined kernel function and any single kernel function. The single kernel functions include Reproducing kernel (Rep KELM) and Polynomial kernel (Poly KELM). Each vector of the dataset has 2 features. 800 training samples and 400 testing samples are randomly selected. Fig. 4 shows the boundaries of different KELM algorithms on benchmark Banana dataset, and compares it with single KELM in Table 2.

<table>
<thead>
<tr>
<th>Training accuracy</th>
<th>Testing accuracy</th>
<th>Testing samples</th>
<th>Identification samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>with no noise 1</td>
<td>1</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>with noise 1</td>
<td>0.96</td>
<td>200</td>
<td>192</td>
</tr>
</tbody>
</table>

Table 1. Classification accuracy of Rep\_P KELM on bi-spiral dataset.
Reproducing Polynomial Kernel Extreme Learning Machine

Fig. 4. Boundaries of different KELM algorithms on Banana dataset.

Table 2. Maximum testing rate [%] on Banana dataset.

<table>
<thead>
<tr>
<th></th>
<th>Kernel Parameters</th>
<th>Weight of the Kernel</th>
<th>Maximum Rate</th>
<th>Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rep</td>
<td>10</td>
<td>0.8</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Poly</td>
<td>10</td>
<td>—</td>
<td>7.1</td>
<td>—</td>
</tr>
<tr>
<td>Rep_P</td>
<td>8</td>
<td>0.7</td>
<td>8.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Table 3. Specifications of multiclass classification cases.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Training data</th>
<th>Testing data</th>
<th>Attributes</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>100</td>
<td>50</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Wine</td>
<td>118</td>
<td>60</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>Vehicle</td>
<td>564</td>
<td>282</td>
<td>18</td>
<td>4</td>
</tr>
<tr>
<td>Waveform</td>
<td>800</td>
<td>400</td>
<td>21</td>
<td>3</td>
</tr>
<tr>
<td>Segment</td>
<td>1540</td>
<td>770</td>
<td>18</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 4. Specifications of regression cases.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Training data</th>
<th>Testing data</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIR</td>
<td>50</td>
<td>10</td>
<td>401</td>
</tr>
<tr>
<td>Autoprice</td>
<td>106</td>
<td>53</td>
<td>15</td>
</tr>
<tr>
<td>Housing</td>
<td>337</td>
<td>169</td>
<td>13</td>
</tr>
<tr>
<td>Bodyfat</td>
<td>168</td>
<td>84</td>
<td>14</td>
</tr>
<tr>
<td>Cloud</td>
<td>682</td>
<td>342</td>
<td>9</td>
</tr>
</tbody>
</table>

6.3. Experiment of Regression on the UCI Benchmark Repository

In order to further compare the performance of Rep_P KELM and other conventional models (Gauss KELM, Gau_P KELM and Rep_P KSVM), 5 multiclass classification cases and 5 regression cases are selected from UCI Machine Learning Repository, they are described in Tables 3 and 4. The classification performance of models are evaluated by maximum testing accuracy and the training time, they are shown in Table 5. The regression performance of models are evaluated by determining coefficient R^2, which is defined within the interval of [0, 1], the closer to 1, the better the regression performance. The simulations results (R^2 and training time) are shown in Table 6. Better test results are given in boldface and the shortest time is in underline.

Tables 5 and 6 show that:

1) All of combined kernel functions have better testing rate in classification cases and have maximum coefficient of determination R^2 in regression cases than single kernel function (Gauss), which Rep_P KELM achieves the best. Besides, due to reproducing kernel has stronger local learning capability, it is observed that Rep_P KELM can always performs better than the traditional combined kernel, Gau_P KELM.

2) The training time of Rep_P KELM is shorter than other models, because Rep_P KELM uses the CS algo-
Table 5. Classification [%] and comparison on standard UCI dataset.

<table>
<thead>
<tr>
<th></th>
<th>Gauss KELM</th>
<th>Gauss_P KELM</th>
<th>Rep_P KSVM</th>
<th>Rep_P KELM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing Rate</td>
<td>Time [s]</td>
<td>(C, σ)</td>
<td>Testing Rate</td>
</tr>
<tr>
<td>Iris</td>
<td>97.2</td>
<td>0.194</td>
<td>(1,1)</td>
<td>98</td>
</tr>
<tr>
<td>Wine</td>
<td>78.3</td>
<td>0.242</td>
<td>(1,4)</td>
<td>90</td>
</tr>
<tr>
<td>Vehicle</td>
<td>76.6</td>
<td>0.416</td>
<td>(1,0,8)</td>
<td>81.5</td>
</tr>
<tr>
<td>Waveform</td>
<td>80.7</td>
<td>2.93</td>
<td>(1,1,4)</td>
<td>84.5</td>
</tr>
<tr>
<td>Segment</td>
<td>87.4</td>
<td>24.8</td>
<td>(10,7)</td>
<td>87.4</td>
</tr>
</tbody>
</table>

Table 6. Regression and comparison on standard UCI dataset.

<table>
<thead>
<tr>
<th></th>
<th>Gauss KELM</th>
<th>Gauss_P KELM</th>
<th>Rep_P KSVM</th>
<th>Rep_P KELM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing Rate</td>
<td>Time [s]</td>
<td>(C, σ)</td>
<td>Testing Rate</td>
</tr>
<tr>
<td>NIR</td>
<td>0.278</td>
<td>0.113</td>
<td>(1,2)</td>
<td>0.824</td>
</tr>
<tr>
<td>Autoprice</td>
<td>0.852</td>
<td>0.115</td>
<td>(1,1,0)</td>
<td>0.866</td>
</tr>
<tr>
<td>Housing</td>
<td>0.866</td>
<td>0.616</td>
<td>(1,3)</td>
<td>0.918</td>
</tr>
<tr>
<td>Bodyfat</td>
<td>0.967</td>
<td>0.117</td>
<td>(1,8)</td>
<td>0.98</td>
</tr>
<tr>
<td>Cloud</td>
<td>0.739</td>
<td>0.930</td>
<td>(10,10)</td>
<td>0.942</td>
</tr>
</tbody>
</table>

Table 7. Comparison of three algorithms on classification dataset.

<table>
<thead>
<tr>
<th></th>
<th>PSO-Rep_P KELM</th>
<th>GA-Rep_P KELM</th>
<th>CS-Rep_P KELM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing Rate</td>
<td>Time [s]</td>
<td>Testing Rate</td>
</tr>
<tr>
<td>Iris</td>
<td>97.7</td>
<td>0.05</td>
<td>93.3</td>
</tr>
<tr>
<td>Wine</td>
<td>88.4</td>
<td>0.63</td>
<td>91.5</td>
</tr>
<tr>
<td>Vehicle</td>
<td>80.6</td>
<td>0.10</td>
<td>81.1</td>
</tr>
<tr>
<td>Waveform</td>
<td>84.8</td>
<td>0.38</td>
<td>85</td>
</tr>
<tr>
<td>Segment</td>
<td>84.6</td>
<td>4.42</td>
<td>85.3</td>
</tr>
</tbody>
</table>

Table 8. Comparison of three algorithms on regression dataset.

<table>
<thead>
<tr>
<th></th>
<th>PSO-Rep_P KELM</th>
<th>GA-Rep_P KELM</th>
<th>CS-Rep_P KELM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Testing Rate</td>
<td>Time [s]</td>
<td>Testing Rate</td>
</tr>
<tr>
<td>NIR</td>
<td>0.933</td>
<td>0.102</td>
<td>0.941</td>
</tr>
<tr>
<td>Autoprice</td>
<td>0.899</td>
<td>0.089</td>
<td>0.899</td>
</tr>
<tr>
<td>Housing</td>
<td>0.918</td>
<td>0.112</td>
<td>0.923</td>
</tr>
<tr>
<td>Bodyfat</td>
<td>0.976</td>
<td>0.256</td>
<td>0.985</td>
</tr>
<tr>
<td>Cloud</td>
<td>0.943</td>
<td>0.051</td>
<td>0.948</td>
</tr>
</tbody>
</table>

Algorithm to optimize the parameters instead of the traditional empirical approach, thus the parameter selection is more efficient and accurate. Especially when compared with Rep_P KSVM, we found Rep_P KSVM achieves almost the same generalization performance as Rep_P KELM, however, the advantage of Rep_P KELM in the training time is very outstanding.

3) The computational complexity of Rep_P KELM is much less than the other models. There are only two parameters in this algorithm, the population size n, and Pa. As long as the initial parameters of the CS algorithm are set, it can quickly and accurately find the appropriate value of (C, b, d, m), reducing the trouble of repeated calculations.

At last, to verify whether the CS algorithm is the superior to the proposed KELM model, this paper chooses some conventional bio-inspired meta heuristic algorithms as a contrast, such as particle swarm optimization (PSO) and genetic algorithm (GA). We still use the datasets in Tables 3 and 4, evaluated by maximum testing accuracy and determining coefficient. The test results are shown in Tables 7 and 8.
Seen from Tables 7 and 8, the CS algorithm can achieve the better testing rate with less time than PSO algorithm and GA algorithm in most regression cases. More importantly, we found that in the PSO algorithm and the GA algorithm, different parameter settings have a greater impact on the final result, so these algorithms need to be adjusted during the process, which increases the computational complexity. In particular, the PSO algorithm also has the problem of easily falling into the local optimal solution. While in CS algorithm, the number of parameters to be tuned is less than PSO and GA algorithm. It is not sensitive to the parameter settings, and the global search ability is stronger and difficult to fall into the local optimal. Therefore, the CS algorithm is more efficient for the proposed model in this paper.

7. Conclusion

Based on the conditions of KELM and reproducing kernel theory, a new combined KELM model built by polynomial kernel and reproducing kernel on Sobolev Hilbert space is proposed. It is strictly proved that the reproducing kernel can be an allowed extreme learning machine kernel function in theory. Evaluations on three different datasets prove that our proposed model performs better than other conventional KELM and KSVM. Besides, CS algorithm has been used to optimize \((C, b, d, m)\), the advantage of proposed model in the training time is very outstanding. This paper has provided a new method for learning KELM and broadens the field of KELM theory, future work will be on the study Rep_P KELM in its practical applications.

Acknowledgements

The project supported by the Natural Science Foundation of Liaoning Province (No.2015020101).

References:


Name: Chao Liu
Affiliation: School of Automation, Shenyang Aerospace University

Brief Biographical History:
2014 Received Bachelor degree from City Institute, Dalian University of Technology
2015- Studied for a Master’s degree in Shenyang Aerospace University
Main Works:
• His main research interests are computational intelligence pattern recognition and artificial intelligence.

Name: Senyue Zhang
Affiliation: College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics

Main Works:
• Her main research interests include Machine Learning and Artificial Intelligence.

Name: Wenan Tan
Affiliation: Nanjing University of Aeronautics and Astronautics

Brief Biographical History:
2003- Professor of Computer Science at Nanjing University of Aeronautics and Astronautics
Main Works:
• His main research interests include software engineering and process engineering.

Name: Yanyan Ding
Affiliation: School of Automation, Shenyang Aerospace University

Brief Biographical History:
2015- Studied for a Master’s degree in Shenyang Aerospace University
Main Works:
• Her main research interests are Machine Learning.