entropy

MDPI

Article

Evolved-Cooperative Correntropy-Based Extreme Learning Machine for Robust Prediction

Wenjuan Mei [1], Zhen Liu [1], Yuanzhang Su [2,*], Li Du [1] and Jianguo Huang [1]

[1] Department of Instrument Science and Technology, University of Electronic Science and Technology of China, Chengdu 611731, China; meiwenjuan@std.uestc.edu.cn (W.M.); scdliu@uestc.edu.cn (Z.L.); summer_christ@163.com (L.D.); xlhjg@uestc.edu.cn (J.H.)

[2] Department of Applied Linguistics, University of Electronic Science and Technology of China, Chengdu 611731, China

* Correspondence: syz@uestc.edu.cn; Tel.: +86-028-6183-0316

Received: 6 August 2019; Accepted: 12 September 2019; Published: 19 September 2019

Abstract: In recent years, the correntropy instead of the mean squared error has been widely taken as a powerful tool for enhancing the robustness against noise and outliers by forming the local similarity measurements. However, most correntropy-based models either have too simple descriptions of the correntropy or require too many parameters to adjust in advance, which is likely to cause poor performance since the correntropy fails to reflect the probability distributions of the signals. Therefore, in this paper, a novel correntropy-based extreme learning machine (ELM) called ECC-ELM has been proposed to provide a more robust training strategy based on the newly developed multi-kernel correntropy with the parameters that are generated using cooperative evolution. To achieve an accurate description of the correntropy, the method adopts a cooperative evolution which optimizes the bandwidths by switching delayed particle swarm optimization (SDPSO) and generates the corresponding influence coefficients that minimizes the minimum integrated error (MIE) to adaptively provide the best solution. The simulated experiments and real-world applications show that cooperative evolution can achieve the optimal solution which provides an accurate description on the probability distribution of the current error in the model. Therefore, the multi-kernel correntropy that is built with the optimal solution results in more robustness against the noise and outliers when training the model, which increases the accuracy of the predictions compared with other methods.

Keywords: correntropy; information theory extreme learning machine; evolved cooperation

1. Introduction

With the rapid development of powerful computing environments and rich data sources, artificial intelligence (AI) technology such as neural networks [1–3], adaptive filtering [4–6] and evolutionary algorithms [7–9] has become increasingly more applicable for forecasting problems in various scenarios, such as medicine [10–12], economy [13–15] and electronic engineering [16–18]. The methods have acquired high reputations due to their great approximation abilities.

Although AI methods perform well when solving real world problems, most corresponding models adapt the mean squared error (MSE) as the criterion for training hidden nodes or building the cost functions, assuming that the data satisfy a Gaussian distribution. Moreover, the MSE is a global similarity measure where all the samples in the joint space have the same contribution [19]. Therefore, the MSE is likely to be badly affected by the noise and outliers that are hiding in the samples and this happens commonly in applications, such as speech signals, images, real-time traffic signals and electronic signals from ill-conditioned devices [20–22]. Therefore, MSE-based models are likely to result in poor performance in real world applications.

To conquer the weaknesses of the least mean squares (LMS), over the past decades, a number of studies have proposed methods to improve the robustness of the model against the noise and outliers that are contained in the data [23–27]. Among the existing technologies, M-estimators have been the focus of many academic studies. By detecting the potential outliers during training procedures, the M-estimator can eliminate the negative influences from the output weights that adversely affect the predictions [28]. Using these advantages, Zhou et al. [29] proposed a novel data-driven standard least-squares support vector regression (LSSVR) applying the M-estimator, which reduces the interference of outliers and enhances the robustness. However, there are difficulties accessing clean learning data without noises so that the application on the M-estimator-based forecasting models based is limited.

Recently, information theoretic learning (ITL) has drawn considerable attention due to its good performance avoiding the effect of the noise and outliers [30–35] and it has become an effective alternative to the MSE criterion. In [36], the authors presented a novel training criterion based on the minimum error entropy (MEE) to replace the MSE. By taking advantages of the higher order description on entropy, MEE has become superior for non-Gaussian signal processing compared with traditional training criteria. Inspired by the entropy and Parzen kernel estimator, Liu et al. [37] proposed an extended definition of the correlation function for random processes using a generalized correlation function, known as correntropy. Although different from global measurements, such as the mean squared error (MSE), the correntropy is regarded as a local similarity measurement where its value is primarily determined by the kernel function along x = y line [38], leading to high robustness against noise and outlies. Moreover, the correntropy has many great properties such as symmetry, nonnegativity and boundness. Most of all, it is easy to form convex cost functions based on the correntropy, which is very convenient for training the models [39–42]. Therefore, the correntropy has been widely used in forming robust models [43–45].

To enhance the forecasting ability of the model, in [46], the correntropy was introduced into the affine projection (AP) algorithm to overcome the degradation of the identification performance with impulsive noise environments. From the simulation results, it is easy to verify that the proposed algorithm has achieved better performance than other methods. Another approach to improve the robustness via the correntropy is enhancing the feature selection efficiencies [47–49]. In [50], the kernel modal regression and gradient-based variable identification were integrated together using the maximum correntropy criterion, which guarantees the robustness of the algorithm. Additionally, in [51], a novel principal component analysis (PCA), based on the correntropy and known as the correntropy-optimized temporal PCA (CTPCA), was adapted to enhance the robustness for rejecting the outlier. The outlier improves the models training in simulation experiments. In addition to providing the extractions of the features in neural networks and filtering methods, the correntropy turns out to be a powerful tool for developing robust training methods that generate and adjust the weights in the model. In [52], Wang et al. introduced a feedback mechanism using the kernel recursive maximum correntropy to provide a novel kernel adaptive filters known as the kernel recursive maximum correntropy with multiple feedback (KRMC-MF). The experiments show that the generated filters have high robustness against outliers. In [53], Ahmad et al. proposed the correntropy based conjugate gradient backpropagation (CCG-BP), which can achieve high robustness in environments with both impulsive noise and heavy-tailed noise distributions. Unfortunately, most of the neural networks have to adjust the weights of each node during each training iterations which wastes time during the training process.

Recently, forecasting models with parameters that are free from adjustments have gained increasingly more attention due to their fast training speeds for the models [54–56]. Combined with the correntropy, these algorithms have shown great potential in real-world applications. For example, Guo et al. [57] developed a novel training method for echo state networks (ESNs) based on a correntropy induced loss function (CLF), which provides robust predictions for time-series signals. Similar to ESNs, extreme learning machines (ELMs) have received great attention on fast learning due to the random assignments of the hidden layer and being equipped with simpler structures, such as single

layer feedback networks (SLFNs) [58–60]. It has been proven that the hidden nodes can be assigned with any continuous probability distribution, while the model satisfies the universal approximation and classification capacity [61]. In particular, the extreme learning machine has been applied and received a high reputation for predicting production processes [62,63], system anomalies [64], etc. [65]. In [66], the authors first developed the correntropy-based ELM that uses the regularized correntropy criterion in place of the MSE with half quadratic (HQ) optimization which is called the regularized correntropy criterion for an extreme learning machine (RCC-ELM). Later, Chen et al. [67] extended the dimensions of the correntropy by combining two kinds of correntropy together to enhance the flexibility of the model to generate more robust ELM called ELM by maximum mixture correntropy criterion (MMCC-ELM). The experimental results show that the learning method performs better than the conventional maximum correntropy method. Although the RCC-ELM and MMCC-ELM possess high robustness compared with other ELM methods, the corresponding correntropy is constrained by no more than two kernels. The kernel bandwidth required for the assignments by users in advance is likely to degrade the model due to the improper description on the probability distribution of the signal with the correntropy.

To conquer the weakness of the existing correntropy-based ELMs, this paper focuses on providing a more robust predicting model with adaptive generation based on multi-kernel correntropy which can bring an accurate description of the current errors of ELM. This study developed a more flexible and robust forecasting ELM based on a newly developed adaptive multi-dimension correntropy using evolving cooperation. In the proposed method, the output weights of the ELM are trained based on the maximum multi-dimension correntropy with no constraints on the dimensions of the kernels. To achieve the most appropriate assignment of the parameters of each kernel in the correntropy, a novel evolving cooperation method is developed to concurrently optimize the bandwidths and the corresponding influence coefficients to achieve the best estimations of the residual errors of the model. Furthermore, the training approach has been developed based on the properties of the multi-dimension correntropy. The main contribution of the paper can be summarized as follows.

- The proposed method develops a novel correntropy criterion with multiple kernels to improve the flexibility for depicting the probability distribution of the current error of the predicting model. Then, a convex cost function has been developed based on the multiple kernel correntropy, which can provide a more robust training strategy for ELMs, resulting in high performance on the predictions against noise and outliers.
- To accurately describe the probability distribution of the current error, the proposed method develops a cooperating evolution strategy to adaptively generate proper bandwidths and coefficients to suit the error distribution which enhances the accuracy on the approximation for the correntropy, leading to more robust training.

The experiments compare the performance of the proposed method and several state-of-art methods using both simulated data and real-world data, which show that the proposed method obtains more the robust predictions than other methods. Finally, the proposed method is incorporated into the forecasting model for the current transfer ratio (CTR) signals for the optical couplers, and it achieves high accuracies and robustness.

The rest of the paper is as follows. The next section introduces the framework of the proposed method and multi-dimension correntropy. Section 3 describes the evolved cooperation for the kernels with multi-dimension correntropy and Section 4 provides the training procedures of the forecasting model. Then, Section 5 estimates the performance of the proposed method using both simulation data and real-world applications. Finally, the conclusion is drawn in Section 6.

2. The Framework of the Proposed Method

The structure of the prediction model that is built using the proposed method is similar to those of other ELM-based methods. Figure 1 shows the basic structure of the method. Generally, the network

includes one input layer, one hidden layer and one output layer. The hidden output is calculated using the given input vectors and the weights and the biases of the hidden nodes which are randomly assigned [54]:

$$\boldsymbol{h} = f(\boldsymbol{w}\boldsymbol{x} + \boldsymbol{b}) \tag{1}$$

where $f(.)$ is the activation function and ($\boldsymbol{w}$,$\boldsymbol{b}$) are the weights and bias of the hidden nodes.

With the hidden layer, the network can simulate any kind of function by generating the output weights with the least mean squares (LMS) The cost function is calculated as follows [58]:

$$\mathrm{J_{LS}} = \|\mathbf{Y} - \mathbf{T}\| \tag{2}$$

where $\mathbf{T}$ is the expected output and $\mathbf{Y}$ is the predicted output of the model. $\mathbf{Y}$ calculated with the hidden outputs h and the output weights $\boldsymbol{\beta}$ as follows:

$$\mathbf{Y} = \boldsymbol{\beta}\boldsymbol{h} \tag{3}$$

Therefore, the output layer is calculated as follows:

$$\boldsymbol{\beta} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{T} \tag{4}$$

Further, to constrain the output weights, the output layer is calculated as follows:

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

where λ is the constraining coefficient.

Although the output weights that are calculated by Equation (4) or Equation (5) can provide good predictions using the training data, the model has suffered with the outliers and noises in the data which negatively affect the predictions. To overcome the problem, the correntropy, as a high order similarity measurement, has been used in some recently developed methods.

In [62], the cost function built using the correntropy as follows:

$$\mathrm{J_{RCC}} = \max_{\beta}[\sum_{\mathrm{p}=1}^{\mathrm{N}} \mathrm{G}(\mathbf{t_p} - \boldsymbol{h}\boldsymbol{\beta}) - \lambda\left\|\boldsymbol{\beta}\right\|] \tag{6}$$

where $\mathrm{G}(\mathbf{t_p} - \boldsymbol{h}\boldsymbol{\beta})$ is the Gussian kernel calculated as follows:

$$\mathrm{G}(\mathbf{t_p} - \boldsymbol{h}\boldsymbol{\beta}) = \exp(-\frac{(\mathbf{t_p} - \boldsymbol{h}\boldsymbol{\beta})}{2\sigma^2}) \tag{7}$$

where σ is the bandwidth of the kernel.

Therefore, the output layer is calculated as follows:

$$\boldsymbol{\beta} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{H} - \lambda\boldsymbol{I})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{T} \tag{8}$$

where $\boldsymbol{\Lambda}$ is the diagonal matrix of the local optimal solution. It is calculated as follows:

$$\alpha_{\mathrm{p}}^{\tau+1} = -\mathrm{G}(\mathbf{t_p} - \boldsymbol{h}\boldsymbol{\beta}) \tag{9}$$

To further improve the flexibility of the correntropy, the cost function with a mixed correntropy is defined in [67] as follows:

$$\mathrm{J_{MMCC}} = 1 - \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}[\alpha\mathrm{G}_{\sigma_1}(\mathrm{e_i}) + (1-\alpha)\mathrm{G}_{\sigma_2}(\mathrm{e_i})] + \lambda\left\|\boldsymbol{\beta}\right\| \tag{10}$$

Therefore, the output is calculated as follows:

$$\boldsymbol{\beta} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{H} + \lambda'\boldsymbol{I})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{T} \tag{11}$$

where the $\lambda' = 2N\lambda$ and $\boldsymbol{\Lambda}$ is the diagonal matrix with elements calculated as follows:

$$\Lambda_{ii} = \alpha/\sigma_1 G_{\sigma_1}(e_i) + (1-\alpha)/\sigma_2 G_{\sigma_2}(e_i) \tag{12}$$

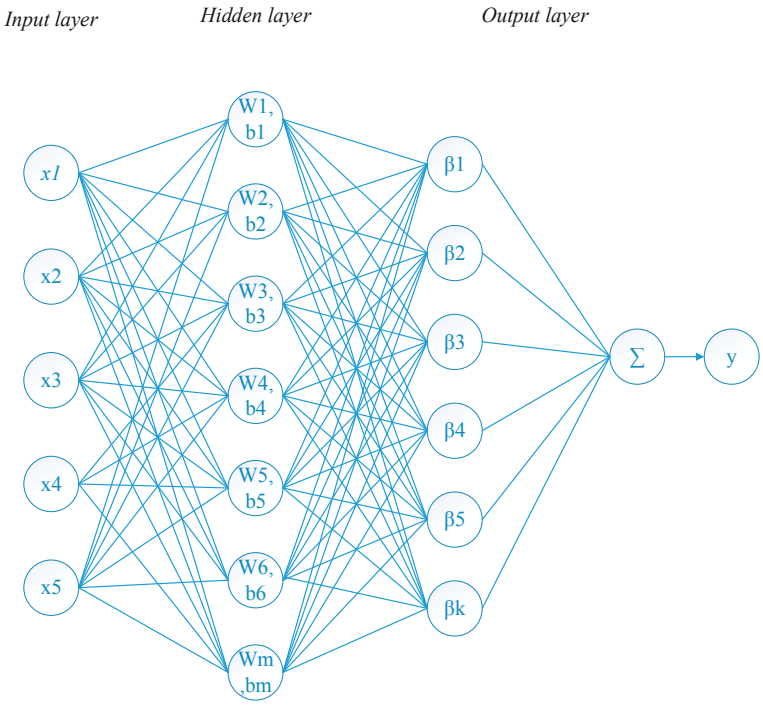


Figure 1. The structure of the prediction model.

With two coefficients, Equation (9) gives a more accurate estimation of the costs of the output layer, leading to a higher robustness of the model. Although Equations (7) and (9) can acquire better local similarity measurements compared with Equation (5), both criterions limit the correntropy into two kernels, leading to an inappropriate description on the probability distribution of the data. Additionally, the bandwidths and the coefficients must be assigned by users, thus limiting the performance of the corresponding model in real world applications which can be badly affected since the bandwidths are not suitable for the estimation of the correntropy. To provide a more flexible criterion for the training strategy with a more appropriate description of the probability distribution of the data, the proposed method develops a multi-kernel correntropy criterion that is calculated as follows:

$$k(\mathbf{T} - \boldsymbol{\beta}\mathbf{H}) = \sum_{i=1}^{K} \alpha_i G_{\sigma_i}(\mathbf{T} - \boldsymbol{\beta}\mathbf{H}) \tag{13}$$

where α_i is the influence coefficients controlling the weight of each kernel. By using multiple kernels to construct the correntropy, the proposed method brings a more accurate approximation on the probability distribution of the samples, leading to a high prediction performance of the model. Based on the corretropy using Equation (13), the proposed method built a convex cost function for training the output weights, which has been analyzed in Section 4. For the suitable assignments of the parameters in Equation (13), a novel generation strategy using an evolved cooperating process based on SDPSO with the MIE to generate the parameters adaptively has been developed. Therefore, the framework of the proposed method can be summarized in Figure 2. The proposed method developed

an evolved-cooperation strategy to generate the optimized solution of the influence coefficients and the bandwidths which suits the distribution of the prediction errors. To achieve an accurate estimation, the bandwidth was generated based on switching delayed particle swarm optimization (SDPSO) [68] and the influence coefficients were calculated based on the cost function for estimating the probability distribution function of errors.

The basic procedures of the method are as follows. Supposing that the input vector of the samples is represented as $\boldsymbol{x} = \{x_1, x_2, \ldots, x_N\}$, calculate the output of hidden nodes with randomly assigned weights and biases as Equation (1). Then, adapt the cooperating evolution technology for training the output weights. For each iterations of the evolution, the output of the predicting model can be generated using Equation (3). Compared with the actual outputs, the predicted outputs result in current error $\boldsymbol{e}$ with the model. Based on the current error $\boldsymbol{e}$, the proposed method makes the best assignments of the bandwidths in the correntropy with SDPSO and accesses the optimal coefficients based on MIE. This is shown in the next section. Using the generated correntropy, a list of diagnostic kernels can be calculated which effects the updating of the output layer to reach higher accuracy. This is presented in Section 4. The processes stop when the cost function of the model is stable.

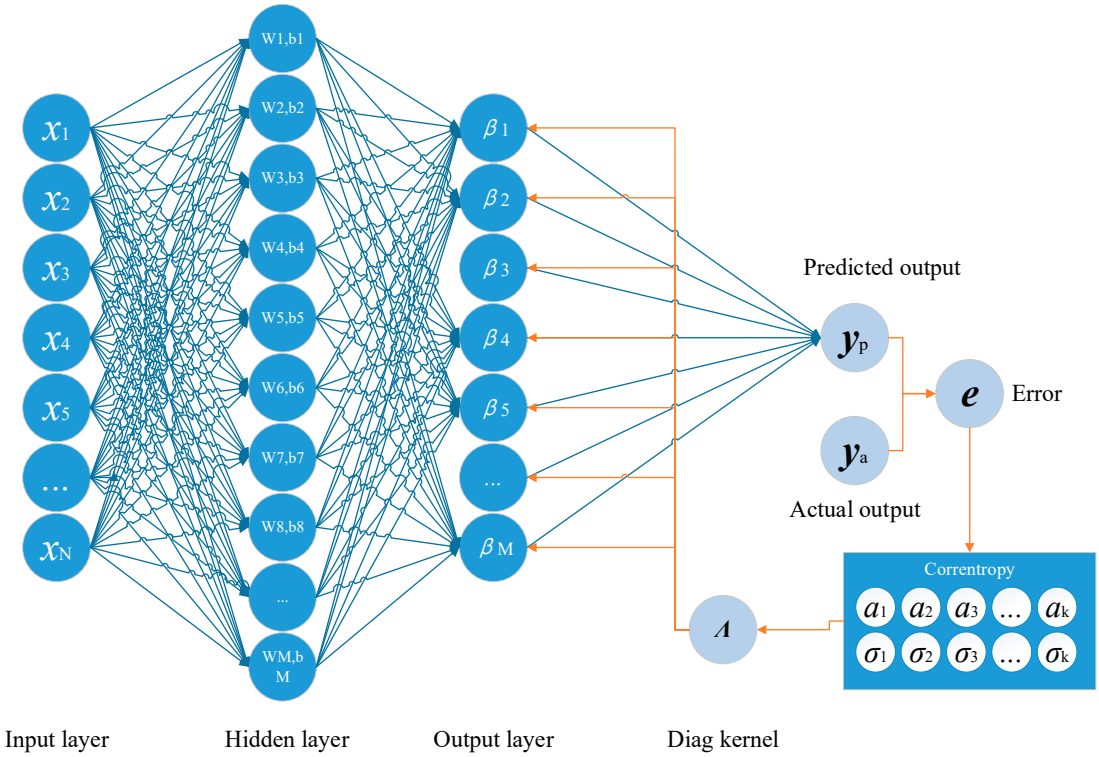


Figure 2. The framework of the proposed method.

More details are presented in the next section.

3. The Cooperating Evolution Process for the Bandwidth and Influence Coefficients of the Kernel

For the correntropy that is defined by Equation (12), the bandwidth and the influence coefficients are for the similarity measurements since the bandwidths act as the zoom lens for the measurements and the coefficients determine the effect that each kernel has on the estimation of the correntropy according to the assigned bandwidth. They are defined as follows:

$$\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_M\} \tag{14}$$

$$\boldsymbol{A} = \{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_M\} \tag{15}$$

Therefore, the bandwidth and the influence coefficients should be carefully assigned to match the probability distribution of the samples to achieve the best effect of the correntropy on generating the output weights of the prediction model. Since the correntropy depicts the probability distribution of the distance between the actual output and the model response, the bandwidth and the coefficients are able to form the probability distribution (pdf) function as follows:

$$\hat{f}(e) = \sum_{i=1}^{N} \alpha_1 G_{\sigma_1}(e_i) + \alpha_2 G_{\sigma_2}(e_i) + \cdots + \alpha_N G_{\sigma_n}(e_i) \tag{16}$$

In applications, the real joint probability distribution for the cases are unknown. Therefore, the joint pdf can only be estimated for a finite number of samples{(ti,yi)}, where i = 1, 2, . . . , N:

$$\mathrm{f(e)} = \frac{1}{\mathrm{N}} \mathrm{g}(\{(\mathrm{t_k}, \mathrm{y_k}) \big| \mathrm{t_k} - \mathrm{y_k} \big| = \mathrm{e}\}) \tag{17}$$

where g(**S**) is the cardinal number of the set **S**.

Using the kernel contrasts between the pdf estimated with the assigned parameters and the pdf estimated using the data, the least mean integrated error (MIE) can be calculated as follows:

$$\mathrm{MIE} = \mathrm{E}(\int (\hat{f}(e) - f(e))^2 de) \tag{18}$$

Based on the MIE, the performance of the bandwidth and coefficients can be estimated using the contrasts with the pdf from the data. Therefore, the optimization of these parameters can be transformed to finding the solution with the minimum MIE.

In the proposed method, the switching delay particle swarm optimization is adapted to search for the best bandwidth. To achieve this, the particles are initialized with a list of potential bandwidth setting $\sigma c = \{\sigma_{c,1}, \sigma_{c,2}, \ldots, \sigma_{c,N}\}$. With respect to each bandwidth of the particle, the velocities for the evolution of the particles are defined as follows:

$$\boldsymbol{v\sigma_c} = \{\mathrm{v}\sigma_{c,1}, \mathrm{v}\sigma_{c,2}, \ldots, \mathrm{v}\sigma_{c,N}\} \tag{19}$$

Meanwhile, the influence coefficient is denoted as vector $\boldsymbol{A}$:

$$\boldsymbol{A_c} = \left\{\alpha_{c,1}, \alpha_{c,2}, \alpha_{c,3}, \ldots, \alpha_{c,M}\right\} \tag{20}$$

where α_i is the influence coefficient according to $\sigma_{c,i.}$

Since the samples provide disperse values of the outputs, the pdf from the data is estimated using the discrete version of Equation (16):

$$\boldsymbol{F} = \{\mathrm{f}(m_1), \mathrm{f}(m_2), \ldots, \mathrm{f}(m_k)\} \tag{21}$$

$$\mathrm{f}(m) = \frac{1}{\mathrm{N}} \mathrm{g}(\{(\mathrm{t_k}, \mathrm{y_k}) \big| m - \varepsilon \le \big| \mathrm{t_k} - \mathrm{y_k} \big| \le m + \varepsilon\}) \tag{22}$$

where the vector $\mathbf{m} = \{m_1, m_2, \ldots, m_k\}$ is a list of values that satisfy $m_1 < m_2 < \ldots < m_k$ and $|m_i - m_{i-1}| = \varepsilon$. ε is the step length of the estimation.

Accordingly, the values from Equation (15) with respect to **m** are equivalent to the following set:

$$\hat{\mathbf{F}} = \left\{\hat{f}(m_1), \hat{f}(m_2), \cdots, \hat{f}(m_k)\right\} \tag{23}$$

They can be calculated as:

$$\hat{\mathbf{F}} = \mathbf{AK} \tag{24}$$

where $\boldsymbol{K}$ is the kernel matrix, which is as follows:

$$K = \begin{bmatrix} G_{\sigma_1}(e_1) & G_{\sigma_1}(e_2) & \cdots & G_{\sigma_1}(e_N) \\ G_{\sigma_2}(e_1) & G_{\sigma_2}(e_2) & \cdots & G_{\sigma_2}(e_N) \\ \vdots & \vdots & \ddots & \vdots \\ G_{\sigma_M}(e_1) & G_{\sigma_M}(e_2) & \cdots & G_{\sigma_M}(e_N) \end{bmatrix} \tag{25}$$

By inserting Equations (20) and (22) into Equation (17), the following cost function can be obtained:

$$\mathrm{MIE} = (\mathbf{AK} - \mathbf{F})(\mathbf{AK} - \mathbf{F})^{\mathrm{T}} \tag{26}$$

Then, the following differential equations with respect to **A** are calculated:

$$2(\mathbf{AK} - \mathbf{F}) = 0 \tag{27}$$

Therefore, the coefficient can be calculated using the assigned bandwidth as follows:

$$\mathbf{A} = \mathbf{F}\mathbf{K}^{T}(\mathbf{K}\mathbf{K}^{T})^{-1} \tag{28}$$

Since each particle contains one solution for the kernels' parameters, the personal best solution $p\sigma$ and the global best solution $g\sigma$ is updated by minimizing the costs. Then, the particles are updated as follows:

$$v\sigma_c(k+1) = wv\sigma_c + c_1(k) \times r_1(p\sigma(k-(k)) - \sigma_c(k))) + c_2(k) \times r_2(g\sigma(k - \tau_2(k) - \sigma_c(k))) \tag{29}$$

$$\sigma_c(k+1) = \sigma_c(k) + v\sigma_c(k+1) \tag{30}$$

where $c_1(k)$ and $c_2(k)$ are the acceleration coefficients and $\tau_1(k)$ and $\tau_2(k)$ are the time delays. All the parameters are adjusted based on the evolution factor, Ef, which determines the evolutionary states, and it is calculated as follows:

$$\mathrm{Ef} = (d_g - d_{\min})/(d_{\max} - d_{\min}) \tag{31}$$

where d_g is the global best particle among the mean distance. It is calculated as:

$$d_g = \frac{1}{N}\sum_{i=1}^{N} \left\| \sigma_{c,i} - g\sigma \right\| \tag{32}$$

With the estimate on Ef, the parameters can be selected as shown in Table 1.

Table 1. The strategies for selecting the parameters.

State	Range of Ef	c_1	c_2	$p\sigma$	$g\sigma$	τ_1	τ_2
Convergence	$0 \le \mathrm{Ef} < 0.25$	2	2	$p\sigma(k)$	$g\sigma(k)$	0	0
Exploitation	$0.25 \le \mathrm{Ef} < 0.5$	2.1	1.9	$p\sigma(k - \tau_1(k))$	$g\sigma(k)$	$[k{\cdot}rand_1]$	0
Exploration	$0.5 \le \mathrm{Ef} < 0.75$	2.2	1.8	$p\sigma(k)$	$g\sigma(k - \tau_2(k))$	0	$[k{\cdot}rand_2]$
Jumping out	$\mathrm{Ef} > 0.75$	1.8	2.2	$p\sigma(k - \tau_1(k))$	$g\sigma(k - \tau_2(k))$	$[k{\cdot}rand_1]$	$[k{\cdot}rand_2]$

The final solution of the bandwidth and the influence coefficients are determined as the solution that minimizes the costs during the evolution procedures.

In summary, the cooperative evolution process is shown in Algorithm 1. First, the bandwidth and the corresponding velocity of each particle are randomly assigned. Then, for each iteration of the process, the influence coefficients are evolved using the bandwidth based on the MIE and the particles are updated using the cost function. Finally, the algorithm finds the best solutions for the

bandwidth and the influence coefficients, from which the kernel depicts the pdf from the data. Based on the generated kernel, the correntropy can lead to a model with good robustness.

Algorithm 1 Evolved cooperation for the kernel parameters

Input: the samples $\{x_i,t_i\}$,i = 1, 2, ... , N
Output: the vector of bandwidth $\boldsymbol{\sigma}$ and the vector of influence coefficients $\boldsymbol{A}$
Parameters: the step length and the number of iterations L
Initialization: Set the cost function of the best solution MIE_{best} to ∞ and randomly assign the bandwidth of the kernels $\boldsymbol{\sigma}_c = \{\sigma_{c,1}, \sigma_{c,2}, \ldots, \sigma_{c,N}\}$ and the corresponding velocity $\mathbf{v}\boldsymbol{\sigma}_c = \{v\sigma_{c,1}, v\sigma_{c,2}, \ldots, v\sigma_{c,N}\}$.
1: **for** k = 1, 2, ... L **do**
2: Generate the best influence coefficients $\boldsymbol{A}$c using Equation (26) for each particles.
3: Calculate value of cost function for each particle MIEc based on Equation (24)
4: Update the personal best solution $p\sigma$ and the global best solution $g\sigma$ based on minimizing the cost function.
5: Calculate the Ef of the iteration with Equation (29)
6: Access the parameters for evolution based on Table 1
7: Update the swarm with Equations (27) and (28)
8: **end for**
9: Return the global best bandwidth $g\sigma$ and the corresponding influence coefficients

4. Training the Extreme Learning Machine Using the Multi-Dimension Correntropy

To improve the robustness of the extreme learning machine, in the proposed method, the training procedure of the output layer as Equation (5), is replaced by the developed calculation using the mixture correntropy that is generated using the evolved kernel from Section 3. The loss function for the output layer is developed according to the following properties.

Property 1. *K(**Y**,**T**) is symmetric, which means the following: K(**Y**,**T**) = K(**T**,**Y**).*

Property 2. *K(**T**,**Y**) is positive and bounded, which means the following: 0 < K(**Y**,**T**) < = 1 and K(**T**,**Y**) = 1 if and only if **T** = **Y**.*

Property 3. *K(**T**,**Y**) involves all the even moments of e, which means the following:*

$$\boldsymbol{K(T,Y)} = \boldsymbol{E[e^{2n}]}\sum_{n=0}^{\infty}\frac{(-1)^n\sum_{i=1}^{M}\alpha_i\sigma_i^{2n}}{2^n\prod_{i=1}^{M}(\sigma_i)^{2n}n!} \tag{33}$$

Property 4. *When the first bandwidth is large enough, it satisfies the following:*

$$\boldsymbol{K(T,Y)} \approx \sum_{i=1}^{M}\alpha_i - \frac{\sum_{i=1}^{M}\alpha_i\sigma_i^2}{2\prod_{i=1}^{M}\sigma_i^2}\boldsymbol{E[e^2]} \tag{34}$$

Proof. For $\lim_{x\to 0}\exp(x) \approx 1 + x$, suppose that σ_1 is large enough, K(**T**,**Y**) can be approximated as follows:

$$\begin{aligned} K(\boldsymbol{T},\boldsymbol{Y}) &= \alpha_1 G_{\sigma_1}(e) + \alpha_2 G_{\sigma_2}(e) + \ldots + \alpha_m G_{\sigma_m}(e) \\ &= \alpha_1\left(1-\frac{e^2}{2\sigma_1^2}\right) + \alpha_2\left(1-\frac{e^2}{2\sigma_2^2}\right) + \ldots + \alpha_m\left(1-\frac{e^2}{2\sigma_m^2}\right) \\ &= \sum_{i=1}^{m}\alpha_i - \frac{\sum_{i=1}^{M}\alpha_i\sigma_i^2}{2\prod_{i=1}^{M}\sigma_i^2}\boldsymbol{E[e^2]} \end{aligned} \tag{35}$$

that completes the proof. □

Remark 1. *Based on Property 4, the mixed C-loss is defined as L(**T**,**Y**) = 1 − K(**T**,**Y**), which is approximately equivalent to the mean square error (MSE) with a large enough bandwidth.*

Property 5. *The empirical mixed C-loss L(e) that is a function of e is convex at any point satisfying* $\|\boldsymbol{e}\|_\infty = \boldsymbol{max}|\boldsymbol{e_i}| \leq \boldsymbol{\sigma_1}$.

Proof. Build the Hessian matrix of the C-loss function L(e) with respect to e as follows:

$$H_{\mathbf{L(e)}}=\left[\frac{\partial \mathbf{L(e)}}{\partial \mathbf{e_i}\partial \mathbf{e_j}}\right]= \mathbf{diag}(\xi_1, \xi_2, \ldots, \xi_N) \tag{36}$$

The elements of matrix ξ, is calculated as follows:

$$\xi_i = \sum_{i=1}^{m} \alpha_i \frac{\sigma_i^4 - e_i^4}{N\sigma_i^4} G_{\sigma_i}(e_i) \tag{37}$$

It is obvious that ξ_i is positive. Therefore, L(e) is convex. □

Remark 2. *Using Property 4 and Property 5, the loss function of the output weights is based on the empirical mixed C-loss L(e) from the data observations, which can be defined as follows:*

$$J = L(\boldsymbol{T}, \boldsymbol{Y}) + \Lambda\left\|\boldsymbol{\beta}\right\|^2 = 1 - \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} \alpha_j G_{\sigma_j}(e_i) + \Lambda\left\|\boldsymbol{\beta}\right\|^2 \tag{38}$$

Based on Equation (38), the training criterion is generated for improvement on the robustness of the model.

Taking the differential of the loss function, it is easy to get the following:

$$\begin{aligned}
&\tfrac{\partial J(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 0 \\
&-\sum_{i=1}^{N} \{[\textstyle\sum_{j=1}^{M} \frac{\alpha_j}{\sigma_j^2} G_{\sigma_j}(e_i)] e_i \boldsymbol{h}_i^T\} + 2N\Lambda\boldsymbol{\beta} = 0 \\
&\sum_{i=1}^{N} (\varphi(e_i)\boldsymbol{h}_i^T\boldsymbol{h}_i\boldsymbol{\beta} - \varphi(e_i)\boldsymbol{t}_i\boldsymbol{h}_i^T) + \Lambda'\boldsymbol{\beta} = 0 \\
&\sum_{i=1}^{N} (\varphi(e_i)\boldsymbol{h}_i^T\boldsymbol{h}_i\boldsymbol{\beta} + \Lambda'\boldsymbol{\beta} = \sum_{i=1}^{N} (\varphi(e_i)\boldsymbol{t}_i\boldsymbol{h}_i^T) \\
&\boldsymbol{\beta} = [\boldsymbol{H}^T\boldsymbol{\Lambda}\boldsymbol{H} + \Lambda'\boldsymbol{I}]^{-1}\boldsymbol{H}^T\boldsymbol{\Lambda}\boldsymbol{T}
\end{aligned} \tag{39}$$

where $\Lambda' = 2N\Lambda$, $\varphi(e_i) = \sum_{j=1}^{M} \frac{\alpha_j}{\sigma_j^2} G_{\sigma_j}(e_i)$ and Λ is a diagonal matrix with diagonal elements $\boldsymbol{\Lambda_{ii}} = \boldsymbol{\varphi}(\boldsymbol{e_i})$, which provides the local similarity measurements between the predicted output and the actual outputs. When the training data contain large noise or many outliers, the corresponding diagonal elements are relatively low which induce the effects of such samples. Therefore, the algorithm can achieve high robustness against noises and outliers in the signals.

Since Equation (37) is a fixed-point equation because the diagonal matrix depends on the weight vector, the optimal solution should be solved by applying the evolved cooperation using Equation (37).

Therefore, combined with the kernel optimization in Section 3, the whole training process can be summarized in Algorithm 2, which is referred to as the ECC-ELM algorithm in this paper.

Algorithm 2 ECC-ELM

Input: the samples $\{x_i, t_i\}$, i = 1, 2, ... , N
Output: output weights
Parameters: the number of hidden nodes N, the number of iterations L, the iterations T and termination tolerance ε
Initialization: Randomly set the weights and bias of the hidden nodes and initialize the output weights $\boldsymbol{\beta}$ using Equation (5)
1: **for** t = 1, 2, ... , T **do**
2: Calculate the residual error: $e_i = t_i - \boldsymbol{h}_i\boldsymbol{\beta}$, i = 1, 2, ... , N
3: Calculate the kernel parameters $\{\boldsymbol{\sigma}, \mathbf{A}\}$ using Algorithm 1
4: Calculate the diagonal matrix $\boldsymbol{\Lambda}$: $\Lambda_{ii} = \varphi(e_i) = \sum_{j=1}^{M} \frac{\alpha_j}{} G_{\sigma_j}(e_i)$
5: Update the output weight using Equation (37)
6: **Until** $|J_k(\boldsymbol{\beta}) - J_{k-1}(\boldsymbol{\beta})| < \varepsilon$
7: **end for**

5. Analysis on Time Complexity and Space Complexity of ECC-ELM

In this section, the time complexity of the proposed method is analyzed and compared with the other algorithms. The main time complexity of the ECCELM comes from the cooperating evolution process and the training process of the model. The cooperative evolution contains the calculations of the influence coefficients and the particles updating with the time complexity of $O(I_t NK^2)$, where I_t is the number of iterations, N is the number of particles and K is the number of disperse values of the outputs. To train the ELM, the procedures share the same time complexity as the RCC-ELM and MMCC-ELM, which is $O(I_h N_l(5M+M^2))$, where I_h is the amount of iterations for training and N_l is the number of training data. Additionally, M is the number of hidden nodes. Therefore, the time complexity of ECC-ELM is $O(I_h N_l(5M+M^2+I_t NK^2))$, which is slightly higher than those of the RCC-ELM and MMCC-ELM but it satisfies the requirements in most applications.

With respect to the spatial complexity, the ECC-ELM has the same complexity as the prediction models using the RCC-ELM, which is $O(N+(N+2)M+N_l{}^2)$. Additionally, the space complexity consumed by evolving process is O(2N+K). Therefore, the space complexity of ECC-ELM is $O(N+(N+2)M+N_l{}^2+2N+K)$, which has the same order as RCC-ELM and MMCC-ELM.

In summary, the time complexity and spatial complexity are practical for most applications.

6. Experiments

6.1. The Simulation of the Sinc Function with Sas noises

In this section, the simulation experiments using the Sinc function with random noises are presented. They compare between serval state-of-art algorithms with the proposed method, which are the R-ELM, the RCC-ELM, the MMCC-ELM and our method. The training and test samples were randomly assigned according to the Sinc function and random noises were added with respect to alpha-stable distribution. This is represented as follows:

$$y = \alpha \mathrm{Sinc}(x) + \rho \tag{40}$$

where α is the scale of the function which is set to 8.0 and Sinc(x) is the Sinc function. The Sinc function is represented as follows:

$$\mathrm{Sinc}(x) = \begin{cases} \sin(x)/x & x \neq 0 \\ 1 & x = 0 \end{cases} \tag{41}$$

Moreover, ρ is the noise that satisfies the following characteristic function [69]:

$$\rho = \begin{cases} \exp\left(-\delta^{\alpha}|\theta|^{\alpha}(1 - j\beta sign(\theta)\tan(\frac{\pi\alpha}{2}))\right) + j\mu\theta & \alpha \neq 1 \\ \exp\left(-\delta^{1}|\theta|^{1}(1 - j\beta(\pi/2)sign(\theta)\log(\frac{\pi\alpha}{2}))\right) & \alpha = 1 \end{cases} \tag{42}$$

The parameters α, β, γ and μ are real and characterize the distribution of the random variable X. Here, the alpha-stable probability distribution function is denoted as $S(\alpha,\beta,\gamma,\mu)$. In these experiments, the four parameters were assigned to three different conditions to provide three types of noises. The assignment of the parameters in each sample is presented in Table 2.

Table 2. The assignments of the parameters in each sample.

Sample #	α	β	γ	μ
Sample 1	1	0	0.001	0
Sample 2	0.7	0	0.0001	0
Sample 3	1.2	0	0.001	0

Each sample contained 200 data, with half of the data being used for training and another half for testing. To get a proper estimation of the performances of each method, the experiments were operated with the best optimization of parameters. This is presented in Table 3.

Table 3. The assignment of the parameters for each algorithms.

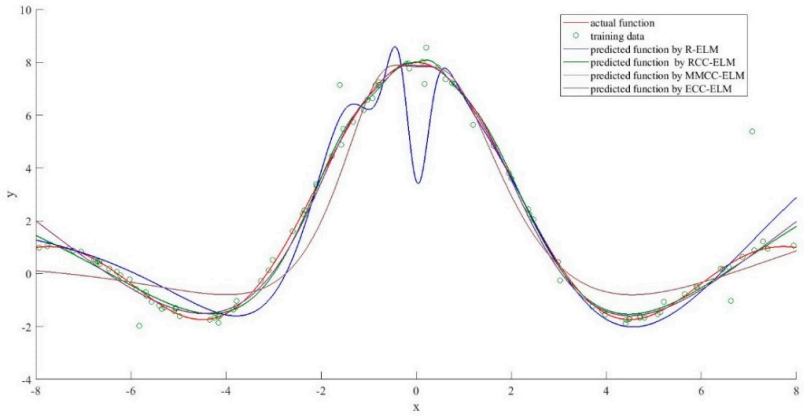
Algorithm	Parameter	Sample 1	Sample 2	Sample 3
R-ELM	N	100	100	100
	λ	0.00001	0.0001	0.0001
RCC-ELM	N	100	100	100
	λ	0.00001	0.00001	0.00001
	I_{hq}	30	30	30
	ε	0.0001	0.0001	0.0001
	σ	1	1.2	1.2
MMCC-ELM	N	100	100	100
	λ	0.00001	0.00001	0.00001
	I_{hq}	30	30	30
	ε	0.0001	0.0001	0.0001
	Σ_1	2	2.2	4.3
	Σ_2	0.8	0.8	8.5
	α	0.8	0.8	0.9
ECC-ELM	N	100	100	100
	λ	0.00001	0.00001	0.00001
	I_{hq}	30	30	30
	ε	0.0001	0.0001	0.0001

Each experiment was conducted 30 times and the averages were taken. The comparison of the accuracies of these algorithms is presented in Table 4. Compared with other algorithms, the R-ELM and ECC-ELM achieve lower mean square errors due to the advantages of the correntropy. The performance of R-ELM is relatively poor due to the effect of noises. The performance of MMCC-ELM also improved by the correntropy. However, since the fixed dimension of the correntropy, the accuracy can be badly influenced by unnecessary assignments on the second order of the bandwidth. Furthermore, it is clear that the proposed algorithm achieves the lowest training MSE, which means that it is the most accurate method for simulation of the Sinc function.

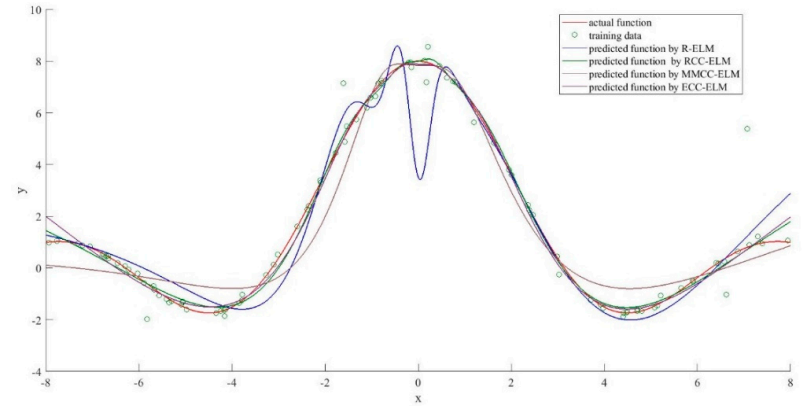
Table 4. The comparison of the accuracies of the four algorithms.

Samples	ELM		RCC-ELM		MMCC-ELM		ECCC-ELM	
	Training MSE	Testing MSE	Training MSE	Testing MSE	Training MSE	Testing MSE	Training MSE	Testing MSE
Sample 1	0.336	0.6601	0.1339	0.3505	0.7225	1.1085	0.1415	0.3595
Sample 2	0.0828	0.11	0.0507	0.0892	1.363	2.189	0.0257	0.0576
Sample 3	0.2219	0.2572	0.2076	0.2339	0.868	0.7583	0.2046	0.2237

To further analyze the predictive abilities of these four algorithms, Figure 3 depicts the differences between the actual function and the predicted function for each algorithm. It is clear that all the algorithms achieve relatively good prediction on the Sinc function. However, the prediction results of the ELM have been badly influenced by the noises in all three samples. Additionally, the MMCC-ELM performance is poor on sample 2 and sample 3, which is probably due to the assignments with high dimension parameters. The RCC-ELM and ECC-ELM provide good predictions, which are almost identical to the actual functions in all three samples. The ECCELM has the closet predicted function with the Sinc function, which also proves that the method has high reliability against noise.

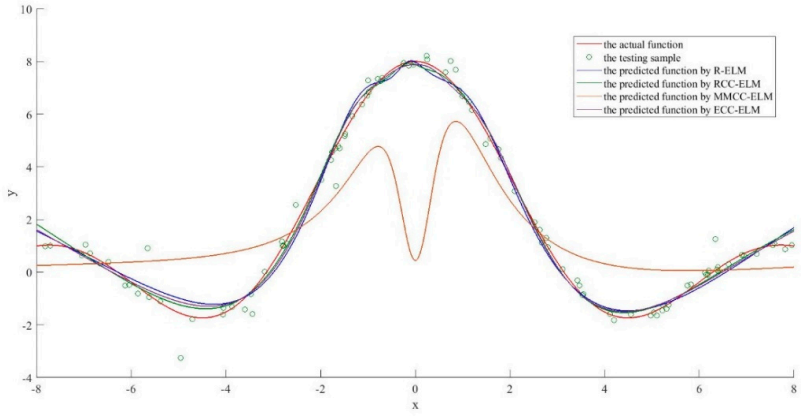


(**a**)



(**b**)

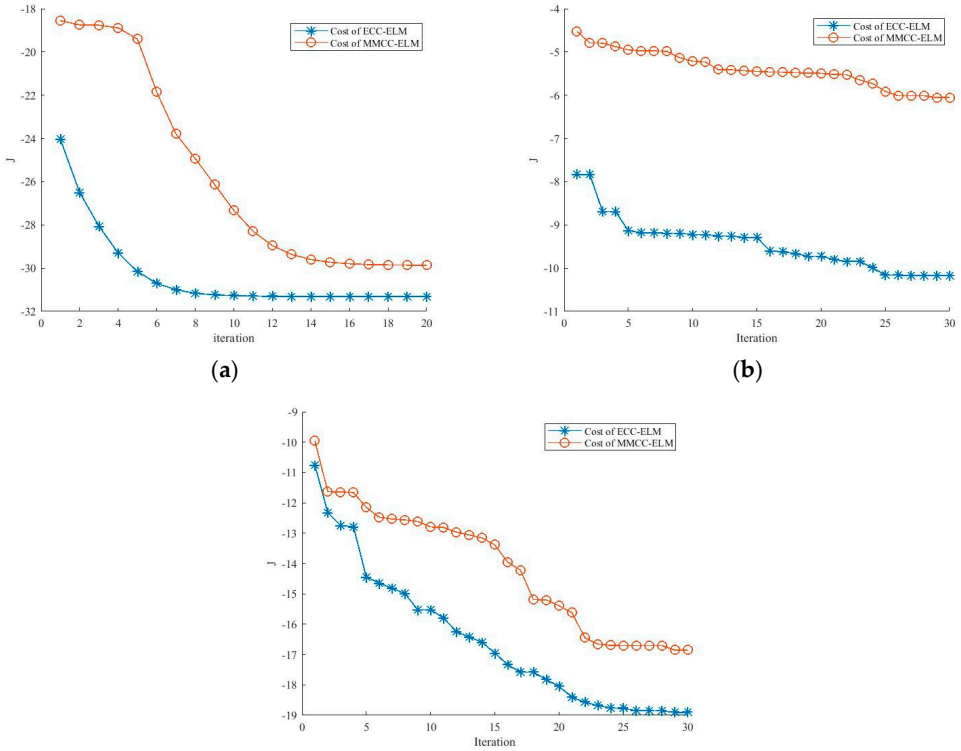
Figure 3. *Cont.*



(**c**)

Figure 3. The performance comparison of each algorithm (**a**) comparison with sample 1; (**b**) comparison with sample 2 and (**c**) comparison with sample 3.

Furthermore, an experiment on sample 1 was conducted to compare the cost function for the output weights with the MMCC-ELM and ECC-ELM since they share similar cost functions. The results are shown in Figure 4, which show that the cost function of ECC-ELM is quite lower than the cost of MMCC-ELM. Additionally, the costs of the ECC-ELM become stable for less than 25 iterations for all three examples than MMCC-ELM. This shows the improvements on training the model with ECC-ELM taking the cooperating evolution technique. Since both algorithms finish the generation of the model when the cost function becomes stable, it can be concluded that the proposed model has faster convergence on training the prediction model.



(**a**) (**b**)

(**c**)

Figure 4. The comparison on the cost function values of the extreme learning machine by maximum mixture correntropy criterion (MMCC-ELM) and ECC-ELM (**a**) comparison with sample 1; (**b**) comparison with sample 2; (**c**) comparison with sample 3.

Figure 5 illustrates the effects of the evolutionary process on the optimization of the kernel bandwidth and influence coefficients. From Figure 5, it can be seen that the cost function for the kernel bandwidth quickly drops during the evolution process. Moreover, Ef continuously decreases during the process, which means that the particle swarm become stable and the best solution occurs. Figure 6 compares the actual pdf function and the estimated pdf function. It can be seen that the algorithm achieves a comparatively accurate estimation of the distribution of the errors.

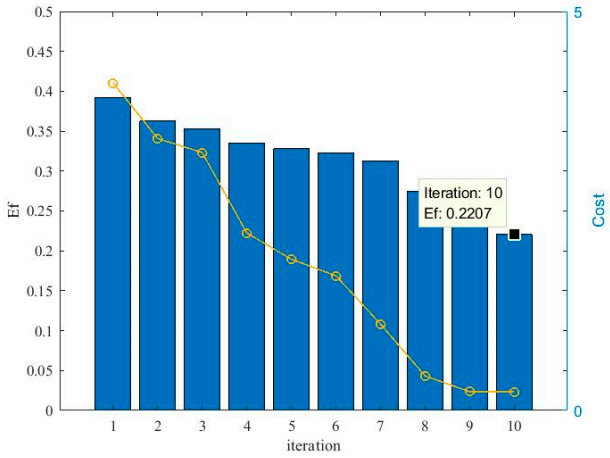


Figure 5. The dynamic changes of the evolution factor (Ef) and costs during the cooperative evolution.

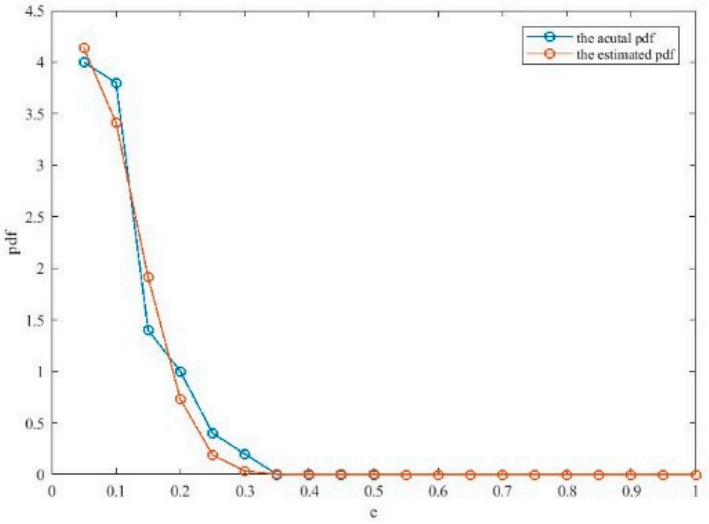


Figure 6. Comparison between the estimated pdf and actual pdf.

6.2. The Performance Comparison on Benchmark datasets

To further assess the proposed algorithm, the performance of the ECC-ELM and other methods were compared using the data set from the UCI machine learning repository [70], awesome public dataset [71] and the United Nations development program [72], which are listed in Table 5. The assignments of the parameters are shown in Table 6, all of which refer to the best performance of each algorithm. Each experiment was conducted 30 times and the average performance was reported.

The performance is compared in Table 7, which shows that the proposed algorithm is able to achieve better prediction accuracies than other methods. Additionally, the performance of the proposed method is relatively stable compared with other correntropy-based extreme learning machines.

Figure 7 compares the actual output value and the predicted value for the Servo data set. It is clear that the predicted values are basically identical to the actual output values, and it has not been influenced by the outliers in the data.

To illustrate the evolutionary processes for optimizing the bandwidth, Figure 8 depicts the distributions of the particles and the evolution of the optimal solutions. It can be seen that the distribution of the particles dynamically changes based on the state of the PSO process. The optimal solution is adjusted and stabilizes during the process, which allows the optimal solution of the bandwidth assignments to generate a more accurate model.

Table 5. The information on the data sets.

Data Set	Features	Observations	
		Training Numbers	Testing Numbers
Servo	5	83	83
Slump	10	52	51
Concrete	9	515	515
Housing	14	253	253
Yacht	6	154	154
Airfoil	5	751	751
Soil moisture	124	340	340
HDI	12	93	93
HIV	10	65	65

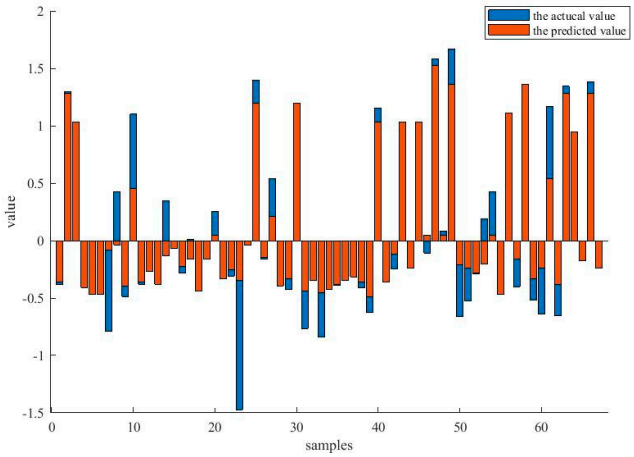


Figure 7. The comparison between the actual values and the predicted values under the data set, Servo.

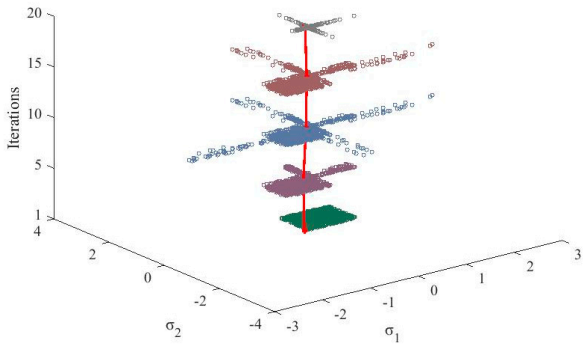


Figure 8. The evolutionary process of the particles.

Table 6. Parameter settings of each algorithm.

Algorithm	Parameter	Servo	Slump	Concrete	Housing	Yacht	Airfoil	Soil Moisture	HDI	HIV
R-ELM	N	90	190	185	180	185	200	200	100	100
	λ	0.00010000	0.00050000	0.00020000	0.00020000	0.00002000	0.00002000	0.00002000	0.00001000	0.00001000
RCC-ELM	N	120	100	200	200	200	180	180	150	120
	λ	0.00001000	0.00010000	0.00000100	0.00010000	0.00000001	0.00000001	0.00000001	0.00000001	0.00000001
	I_{hq}	30	30	30	30	30	30	30	30	30
	ε	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000
	σ	0.00100000	0.00001000	0.00005000	0.01000000	0.00000100	0.00000100	0.00000100	0.00000100	0.00000130
MMCC-ELM	N	90	165	200	200	195	150	150	150	150
	λ	0.00100000	0.00001000	0.00005000	0.01000000	0.00000100	0.00000100	0.00000100	0.00000100	0.00000100
	I_{hq}	30	30	30	30	30	30	30	30	30
	ε	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000
	Σ_1	0.2	0.5	0.5	0.5	0.5	0.2	1.0	1.2	0.7
	Σ_2	2.8	1.6	2.6	2	2	2.7	0.7	0.8	0.3
	α	0.8	0.3	0.5	0.8	0.8	0.5	0.6	0.7	0.6
ECC-ELM	N	90	180	180	180	180	200	200	200	200
	λ	0.00100000	0.00001000	0.00005000	0.01000000	0.00000100	0.00000100	0.00000100	0.00000100	0.00000100
	I_{hq}	30	30	30	30	30	30	30	30	30
	ε	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000	0.00010000

Table 7. The performance comparison.

Data Set	R-ELM		RCC-ELM		MMCC-ELM		ECCC-ELM	
	Training RMSE	Testing RMSE	Training RMSE	Testing RMSE	Training RMSE	Testing RMSE	Training RMSE	Testing RMSE
Servo	0.0590 ± 0.009	0.1039 ± 0.0164	0.0740 ± 0.0106	0.1031 ± 0.0148	0.0839 ± 0.0174	0.0989 ± 0.0187	0.1047 ± 0.0181	0.8742 ± 0.0131
Slump	0.0081 ± 0.0011	0.0461 ± 0.0095	0.0000 ± 0.0000	0.0422 ± 0.0094	0.0001 ± 0.0000	0.0408 ± 0.0101	0.0001 ± 0.0001	0.354 ± 0.1890
Concrete	0.0738 ± 0.0021	0.0917 ± 0.0045	0.0561 ± 0.0018	0.0872 ± 0.0066	0.0560 ± 0.0021	0.0867 ± 0.0064	0.0561 ± 0.0018	0.0852 ± 0.0053
Housing	0.0439 ± 0.0043	0.0896 ± 0.0124	0.0495 ± 0.0045	0.0830 ± 0.0110	0.0554 ± 0.0045	0.0821 ± 0.0101	0.0352 ± 0.0013	0.0791 ± 0.0110
Yacht	0.0366 ± 0.0093	0.0529 ± 0.0090	0.0125 ± 0.0008	0.0349 ± 0.0113	0.0125 ± 0.0008	0.0328 ± 0.0074	0.0172 ± 0.0027	0.0268 ± 0.0031
Airfoil	0.0974 ± 0.0074	0.1031 ± 0.0077	0.0736 ± 0.0022	0.0906 ± 0.0054	0.0736 ± 0.0025	0.0898 ± 0.0051	0.0736 ± 0.0023	0.0889 ± 0.0046
Soil moisture	0.0032 ± 0.0011	0.0095 ± 0.0013	0.0007 ± 0.0001	0.0015 ± 0.0003	0.0006 ± 0.0000	0.0012 ± 0.0002	0.0006 ± 0.0000	0.0009 ± 0.0001
HDI	0.0004 ± 0.0001	0.0006 ± 0.0002	0.0001 ± 0.0000	0.0003 ± 0.0001	0.0001 ± 0.0000	0.0003 ± 0.0001	0.0001 ± 0.0000	0.0003 ± 0.0001
HIV	0.0376 ± 0.0220	0.0599 ± 0.0130	0.0050 ± 0.0017	0.0079 ± 0.0009	0.0047 ± 0.0006	0.0065 ± 0.0004	0.0059 ± 0.0007	0.0059 ± 0.0006

6.3. The Performance Estimations for Forecasting the CTR of Optical Couplers

Finally, to estimate the performance of a real application, the proposed method has been used to predict the current transfer ratio for optical couplers. This is one type of transmission device for electric signals and optical signals with wide applications to the isolation transfer of signals, A/D transmission, D/A transmission, digital communications and high-pressure control. For optical couplers, the CTR is an essential factor for estimating the operating status of optical couplers. In this section, the proposed method was used to give the predictions of CTR for the optical couplers to predict the health condition of the devices.

For the experiments, the degenerating signals of four optical couplers were recorded and transformed into the samples historical CTR value as input vectors and the CTR value of the next time as the expected output. The training data was the samples that were generated from the optical couplers' records over the first ten years and the testing data were the samples that were generated from the last ten years.

Figure 9 depicts the evolutionary process of the PSO procedure. It shows that the Ef value quickly decreases during the evolutionary process and stabilizes within 17 iterations, resulting in the optimal solution that is provided by the swarm.

Finally, the predicted results of the four optical couplers are shown in Figure 10. It is clear that the generated ELM network accurately predicts the CTR value of each optical coupler and is robust with the noises of the signals. Therefore, the proposed method is able to achieve good performance for the optical couplers.

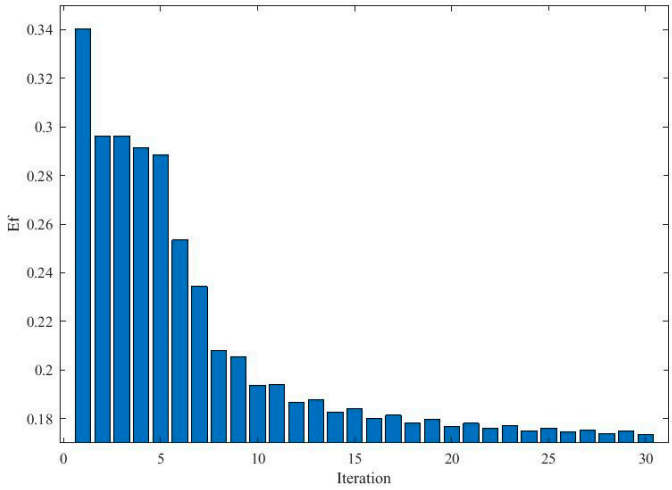


Figure 9. Dynamic changes on Ef and costs during the cooperative evolution.

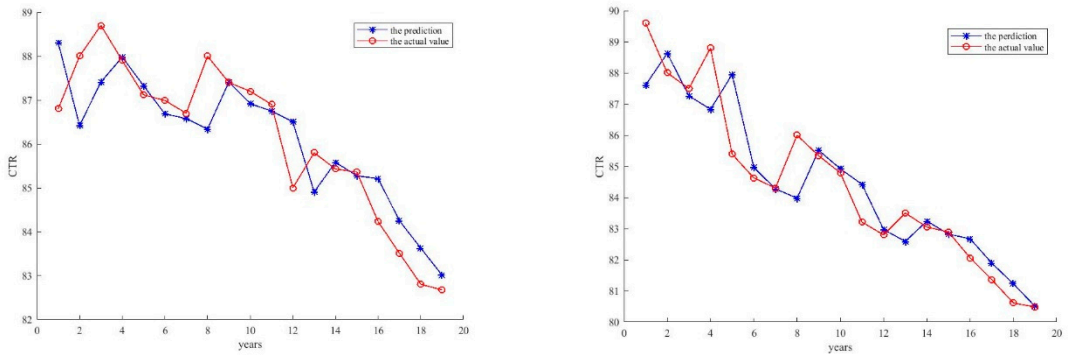


Figure 10. *Cont.*

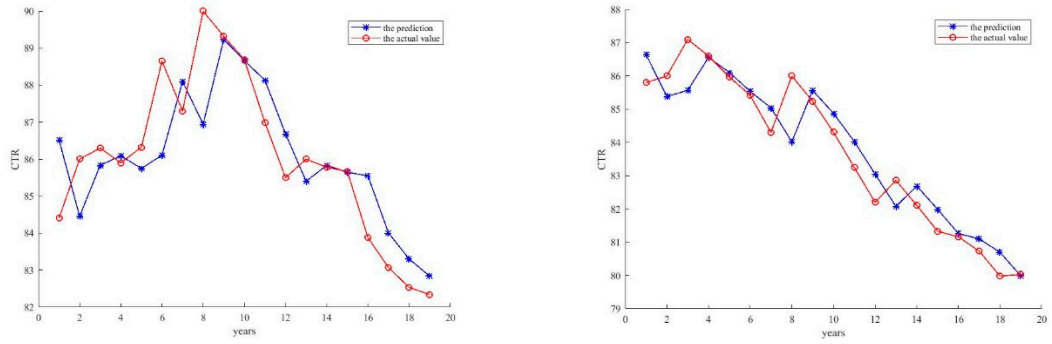


Figure 10. The comparison with the predicted current transfer ratio (CTR) and actual CTR.

Table 8 presents the numerical results of the CTR prediction, which compares the actual CTR and the predicted CTR. It is clear that the proposed method can very accurately provide the prediction on the state of Optical Couplers (OCs). Additionally, the time consumption is presented in Table 8 which shows that the proposed method is able to obtain high accuracy on the prediction of the future CTR of the OC and the predicting time is quite low within 5 ms. Therefore, the proposed method can achieve high performance on real applications.

Table 8. The performance of the predicted model that is generated using the ECCELM.

Time (year)	Actual CTR	Predicted CTR	Normalized Error	Predicting Time (ms)
1	87.90	88.03	0.0037	2.98
2	87.70	88.01	0.0068	4.02
3	87.40	87.94	0.0274	3.92
4	85.50	87.15	0.0095	4.98
5	86.30	87.02	0.0122	2.26
6	85.93	86.61	0.0084	3.22
7	85.86	85.40	0.0188	5.74
8	84.73	85.30	0.0145	4.48
9	84.01	84.33	0.0115	5.85
10	83.31	83.38	0.0023	4.87

7. Conclusions

To improve the robustness of the forecasting model, the paper provides a novel correntropy-based ELM called the ECC-ELM. It uses a multi-dimension correntropy criterion and the evolved cooperation method to adaptively generate the parameters for kernels. In the proposed algorithm, SDPSO is integrated by minimizing the MIE to determine the proper bandwidths and their corresponding influence coefficients to estimate the probability distributions of the residual error of the model. A novel training process was developed based on the properties of the multi-dimension correntropy and it was able to build the convex cost function to calculate the output weights for the ELM. The experiments on the simulated data and real-world application were conducted to estimate the accuracy of the probability distribution of the signal and robustness on predicting the samples. The simulation results with the Sinc function proved that the proposed method can generate the multi-kernel correntropy with high accuracy on describing the probability distribution of the signals and fast converge on the evolution process. This leads to high robustness of the proposed method compared with the other methods. The performance comparisons on the benchmark datasets show that the proposed method can achieve higher accuracy and more stability than the other methods. Finally, the CTR prediction experiments show the proposed method can achieve high accuracy within acceptable time consumption on real world applications. Although the proposed algorithm has predictive advantages, there are still several limitations on the study. One limitation is the proposed method is only applicable for an ELM with one hidden layer, which requires extensions on multi-layer networks. The other

limitation is that the proposed method only provides an offline training model. Therefore, how to update the online prediction model becomes another interesting topic for future research. The codes and data of the research are available at https://github.com/mwj1997/ECC-ELM.

Author Contributions: Conceptualization, W.M.; Data curation, L.D.; Funding acquisition, Z.L.; Investigation, J.H.; Methodology, W.M. and Y.S.; Project administration, Z.L.; Resources, J.H.; Software, W.M.; Supervision, Z.L.; Validation, Y.S.; Writing—original draft, W.M.; Writing—review & editing, L.D.

Funding: This work was supported by the National Natural Science Foundation of China under Grants No. U1830133 (NSAF) and No. 61271035.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Heddam, S.; Keshtegar, B.; Kisi, O. Predicting total dissolved gas concentration on a daily scale using kriging interpolation, response surface method and artificial neural network: Case study of Columbia river Basin Dams, USA. *Nat. Resour. Res.* **2019**, *2*, 1–18. [CrossRef]
2. Ahmadi, N.; Nilashi, M.; Samad, S.; Rashid, T.A.; Admadi, H. An intelligent method for iris recognition using supervised machine learning techniques. *Opt. Laser Thechnol.* **2019**, *120*, 105701. [CrossRef]
3. Aeukumar, R.; Karthigaikumar, P. Multi-retinal disease classification by reduced deep learning features. *Neural Comput. Appl.* **2017**, *28*, 329–334.
4. Pentapati, H.K.; Teneti, M. Robust speaker recognition systems with adaptive filter algorithms in real time under noisy conditions. *Adv. Decis. Sci. Image Process. Secur. Comput. Vis.* **2020**, *4*, 1–18.
5. Eweda, E. Stability bound of the initial mean-square division of high-order stochastic gradient adaptive filtering algorithms. *IEEE Trans. Signal Process.* **2019**, *6*, 4168–4176. [CrossRef]
6. Huang, X.; Wen, G.; Liangm, L.; Zhang, Z.; Tan, Y. Frequency phase space empirical wavelet transform for rolling bearing fault diagnosis. *IEEE Access.* **2019**, *7*, 86306–86318. [CrossRef]
7. Yang, J.; Zhu, H.; Liu, T. Secure and economical multi-cloud storage policy with NSGA-II-C. *Appl. Soft Comput.* **2019**, *83*, 105649. [CrossRef]
8. Albasri, A.; Abdali-Mohammadi, F.; Fathi, A. EEG electrode selection for person identification thru a genetic-algorithm method. *J. Med. Syst.* **2019**, *43*, 297. [CrossRef]
9. Dermanaki Farahani, Z.; Ahmadi, M.; Sharifi, M. History matching and uncertainty quantification for velocity dependent relative permeability parameters in a gas condensate reservoir. *Arab. J. Geosci.* **2019**, *12*, 454. [CrossRef]
10. Shah, P.; Kendall, F.; Khozin, S.; Goosen, R.; Hu, J.; Laramine, J.; Ringel, M.; Schork, N. Artificial intelligence and machine learning in clinical development: A translational perspective. *Nature* **2019**, *2*, 1–5. [CrossRef]
11. Shirwaikar, R.D.; Dinesh, A.U.; Makkithaya, K.; Suruliverlrajan, M.; Srivastava, S.; Leslie, E.S.; Lewis, U. Optimizing neural network for medical data sets: A case study on neonatal apnea prediction. *Artif. Intell. Med.* **2019**, *98*, 59–76. [CrossRef] [PubMed]
12. Lucena, O.; Souza, R.; Rittner, L.; Frayne, R.; Lotufo, R. Convolutional neural network for skull-stripping in brain MR imaging using silver standard masks. *Artif. Intell. Med.* **2019**, *98*, 48–58. [CrossRef] [PubMed]
13. Guan, H.; Dai, Z.; Guan, S.; Zhao, A. A neutrosophic forecasting model for time series based on first-order state and information entropy of high-order fluctuation. *Entropy* **2019**, *21*, 455. [CrossRef]
14. Tymoshchuk, O.; Kirik, O.; Dorundiak, K. Comparative analysis of the methods for assessing the probability of bankruptcy for Ukrainian enterprises. In *Lecture Notes in Computational Intelligence and Decision Making*; Springer: Basel, Switzerland, 2019; pp. 281–293.
15. Yang, T.; Jia, S. Research on artificial intelligence technology in computer network technology, International conference on artificial intelligence and security. In Proceedings of the 5th International Conference on Artificial Intelligence and Security (ICAIS 2019), New York, NY, USA, 26–28 July 2019; pp. 488–496.
16. Senguta, E.; Jain, N.; Garg, D.; Choudhury, T. A review of payment card fraud detection methods using artificial intelligence. In Proceedings of the International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), Belagavi, India, 21–23 December 2018; pp. 494–499.
17. Ampatzidis, Y.; Partel, V.; Meyering, B.; Albercht, U. Citrus rootstock evaluation utilizing UAV-based remote sensing and artificial intelligence. *Comput. Electron. Agric.* **2019**, *164*, 104900. [CrossRef]

18. Yue, D.; Han, Q. Guest editorial special issue on new trends in energy internet: Artificial intelligence-based control, network security and management. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1551–1553. [CrossRef]
19. Liu, W.; Pokharel, P.P.; Principe, J.C. The kernel least mean square algorithm. *IEEE Trans. Signal Process.* **2008**, *56*, 543–554. [CrossRef]
20. Vega, L.R.; Rey, H.; Benesty, J.; Tressens, S. A new robust variable step-size NLMS algorithm. *IEEE Trans. Signal Process.* **2008**, *56*, 1878–1893. [CrossRef]
21. Vega, L.R.; Rey, H.; Benesty, J.; Tressens, S. A fast robust recursive least-squares algorithm. *IEEE Trans. Signal Process.* **2008**, *57*, 1209–1216. [CrossRef]
22. Ekpenyong, U.E.; Zhang, J.; Xia, X. An improved robust model for generator maintenance scheduling. *Electr. Power Syst. Res.* **2012**, *92*, 29–36. [CrossRef]
23. Huang, Y.; Lee, M.-C.; Tseng, V.S.; Hsiao, C.; Huang, C. Robust sensor-based human activity recognition with snippet consensus neural networks. In Proceedings of the IEEE 16th International Conference on Wearable and Implantable Body Sensor Networks (BSN), Chicago, IL, USA, 19–22 May 2019.
24. Ning, C.; You, F. Deciphering latent uncertainty sources with principal component analysis for adaptive robust optimization. *Comput. Aided Chem. Eng.* **2019**, *46*, 1189–1194.
25. He, C.; Zhang, Q.; Tang, Y.; Liu, S.; Liu, H. Network embedding using semi-supervised kernel nonnegative matrix factorization. *IEEE Access.* **2019**, *7*, 92732–92744. [CrossRef]
26. Bravo-Moncayo, L.; Lucio-Naranjo, J.; Chavez, M.; Pavon-Garcia, I.; Garzon, C. A machine learning approach for traffic-noise annoyance assessment. *Appl. Acoust.* **2019**, *156*, 262–270. [CrossRef]
27. Santos, J.D.A.; Barreto, G.A. An outlier-robust kernel RLS algorithm for nonlinear system identification. *Nonlinear Dyn.* **2017**, *90*, 1707–1726. [CrossRef]
28. Guo, W.; Xu, T.; Tang, K. M-estimator-based online sequential extreme learning machine for predicting chaotic time series with outliers. *Neural Comput. Appl.* **2017**, *28*, 4093–4110. [CrossRef]
29. Zhou, P.; Guo, D.; Wang, H.; Chai, T. Data-driven robust M-LS-SVR-based NARX modeling for estimation and control of molten iron quality indices in blast furnace ironmaking. *IEEE Trans. Neural Netw. Learn.* **2018**, *29*, 4007–4021. [CrossRef]
30. Ma, W.; Qiu, J.; Liu, X.; Xiao, G.; Duan, J.; Chen, B. Unscented Kalman filter with generalized correntropy loss for robust power system forecasting-aided state estimation. *IEEE Trans. Ind. Inf.* **2019**. [CrossRef]
31. Safarian, C.; Ogunfunmi, T. The quaternion minimum error entropy algorithm with fiducial point for nonlinear adaptive systems. *Signal Process.* **2019**, *163*, 188–200. [CrossRef]
32. Dighe, P.; Asaei, A.; Bourlard, H. Low-rank and sparse subspace modeling of speech for DNN based acoustic modeling. *Speech Commun.* **2019**, *109*, 34–45. [CrossRef]
33. Li, L.-Q.; Wang, X.-L.; Xie, W.-X.; Liu, Z.-X. A novel recursive T-S fuzzy semantic modeling approach for discrete state-space systems. *Neurocomputing* **2019**, *340*, 222–232. [CrossRef]
34. Hajiabadi, M.; Hodtani, G.A.; Khoshbin, H. Robust learning over multi task adaptive networks with wireless communication links. *IEEE Trans. Comput. Aided Des.* **2019**, *66*, 1083–1087.
35. Kutz, N.J. Neurosensory network functionality and data-driven control. *Curr. Opin. Syst. Biol.* **2019**, *3*, 31–36. [CrossRef]
36. Chen, B.; Xing, L.; Zheng, N.; Principe, J.C. Quantized minimum error Entropy criterion. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 1370–1380. [CrossRef]
37. Liu, W.; Pokharel, P.P.; Principe, J.C. Correntropy: Properties and applications in non-guassian signal processing. *IEEE Trans. Signal Process.* **2007**, *55*, 5286–5298. [CrossRef]
38. Kuliova, M.V. Factor-form Kalman-like implementations under maximum correntropy criterion. *Signal Process.* **2019**, *160*, 328–338. [CrossRef]
39. Ou, W.; Gou, J.; Zhou, Q.; Ge, S.; Long, F. Discriminative Multiview nonnegative matrix factorization for classification. *IEEE Access.* **2019**, *7*, 60947–60956. [CrossRef]
40. Wang, Y.; Yang, L.; Ren, Q. A robust classification framework with mixture correntropy. *Inform. Sci.* **2019**, *491*, 306–318. [CrossRef]
41. Moustafa, N.; Turnbull, B.; Raymond, K. An ensemble intrusion detection technique based on proposed statical flow features for protecting network traffic of internet of things. *IEEE Internet Things J.* **2019**, *6*, 4815–4830. [CrossRef]

42. Wang, G.; Zhang, Y.; Wang, X. Iterated maximum correntropy unscented Kalman filters for non-Gaussian systems. *Signal Process.* **2019**, *163*, 87–94. [CrossRef]
43. Peng, J.; Li, L.; Tang, Y.Y. Maximum likelihood estimation-based joint sparse representation for the classification of hyperspectral remote sensing images. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 1790–1802. [CrossRef]
44. Masuyama, N.; Loo, C.K.; Wermter, S. A kernel Bayesian adaptive resonance theory with a topological structure. *Int. J. Neural Syst.* **2019**, *29*, 1850052. [CrossRef]
45. Shi, W.; Li, Y.; Wang, Y. Noise-free maximum correntropy criterion algorithm in non-Gaussian environment. *IEEE Trans. Circuits Syst. II Express Briefs* **2019**. [CrossRef]
46. Jiang, Z.; Li, Y.; Hunag, X. A correntropy-based proportionate affine projection algorithm for estimating sparse channels with impulsive noise. *Entropy* **2019**, *21*, 555. [CrossRef]
47. He, R.; Zheng, W.-S.; Hu, B.-G. Maximum correntropy criterion for robust face recognition. *IEEE Trans. Patt. Anal. Mach. Intell.* **2019**, *33*, 1561–1576.
48. Macheshwari, S.; Pachori, R.B.; Rajendra, U. Automated diagnosis of glaucoma using empirical wavelet transform and correntropy features extracted from fundus images. *IEEE J. Biol. Health Inf.* **2017**, *21*, 803–813. [CrossRef]
49. Moharmmadi, M.; Noghabi, H.S.; Hodtani, G.A.; Mashhadi, H.R. Robust and stable gene selection via maximum minimum correntropy criterion. *Geomics* **2016**, *107*, 83–87.
50. Guo, C.; Song, B.; Wang, Y.; Chen, H.; Xiong, H. Robust variable selection and estimation based on modal regression. *Entropy* **2019**, *21*, 403. [CrossRef]
51. Luo, X.; Xu, Y.; Wang, W.; Yuan, M.; Ban, X.; Zhu, Y.; Zhao, W. Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy. *J. Frankl. Inst.* **2018**, *355*, 1945–1966. [CrossRef]
52. Wang, S.; Dang, L.; Wang, W.; Qian, G.; Chi, K.T.S.E. Kernel adaptive filters with feedback based on maximum correntropy. *IEEE Access.* **2018**, *6*, 10540–10552. [CrossRef]
53. Heravi, A.R.; Hodtani, G.A. A new correntropy-based conjugate gradient backpropagation algorithm for improving training in neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 6252–6263. [CrossRef]
54. Jaeger, H.; Lukosevicious, M.; Popovivi, D.; Siewert, U. Optimization and applications of echo state networks with leaky integrator neurons. *Neural Netw.* **2007**, *20*, 335–352. [CrossRef]
55. Tanaka, G.; Yamane, T.; Heroux, J.B.; Nakane, R.; Kanazawa, N.; Takeda, S.; Numata, H.; Nakano, D.; Hirose, A. Recent advances in physical reservoir computing: A review. *Neural Netw.* **2019**, *115*, 100–123. [CrossRef]
56. Obst, O.; Trinchi, A.; Hardin, S.G.; Chawick, M.; Cole, I.; Muster, T.H.; Hoschke, N.; Ostry, D.; Price, D.; Pham, K.N. Nano-scale reservoir computing. *Nano Commun. Netw.* **2013**, *4*, 189–196. [CrossRef]
57. Guo, Y.; Wang, F.; Chen, B.; Xin, J. Robust echo state network based on correntropy induced loss function. *Neurocomputing* **2017**, *267*, 295–303. [CrossRef]
58. Huang, G.; Chen, L. Convex incremental extreme learning machine. *Neurocomputing* **2007**, *70*, 3056–3062. [CrossRef]
59. Huang, G.; Zhou, H.; Ding, X.; Zhang, R. Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst. Man Cybern. Part B* **2012**, *42*, 513–529. [CrossRef]
60. Tang, J.; Deng, C.; Huang, G. Extreme learning machine for multilayer perceptron. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 809–821. [CrossRef]
61. Huang, G.; Bai, Z.; Lekamalage, L.; Vong, C.M. Local receptive fields based extreme learning machine. *IEEE Comput. Intell. Mag.* **2015**, *10*, 18–29. [CrossRef]
62. Arabilli, S.F.; Najafi, B.; Alizamir, M.; Mosavi, A.; Shamshirband, S.; Rabczuk, T. Using SVM-RSM and ELM-RSM Approaches for optimizing the production process of Methyl and Ethyl Esters. *Energies* **2018**, *11*, 2889.
63. Ghazvinei, P.T.; Darvishi, H.H.; Mosavi, A.; Yusof, K.b.W.; Alizamir, M.; Shamshirband, S.; Chau, K.-W. Sugarcane growrh prediction based on meteorological parameter using extreme learning machine and artificial neural network. *Eng. Appl. Comp. Fluid.* **2018**, *12*, 738–749.
64. Shamshirband, S.; Chronopoulos, A.T. A new malware delectation system using a high performance ELM method. In Proceedings of the 23rd international database applications & engineering symposium, Athens, Greece, 10–12 June 2019; p. 33.

65. Bin, G.; Yan, X.; Yang, X.; Gary, W.; Shuyong, L. An intelligent time-adaptive data-driven method for sensor fault diagnosis in induction motor drive system. *IEEE Trans. Ind. Electr.* **2019**, *66*, 9817–9827.
66. Xing, H.; Wang, X. Training extreme learning machine via regularized correntropy criterion. *Neural Comput. Appl.* **2013**, *23*, 1977–1986. [CrossRef]
67. Chen, B.; Wang, X.; Lu, N.; Wang, S.; Cao, J.; Qin, J. Mixture correntropy for robust learning. *Pattern Recognit.* **2018**, *79*, 318–327. [CrossRef]
68. Zeng, N.; Zhang, H.; Liu, W.; Liang, J.; Alsaadi, F.E. A switching delayed PSO optimized extreme learning machine for short-term load forecasting. *Neurocomputing* **2017**, *240*, 175–182. [CrossRef]
69. Weron, A.; Weron, R. Computer simularion of Levy alpha-stable variables and processes. In *Lecture Notes in Pihysics*; Springer: Berlin/Heidelberg, Germany, 1995; pp. 379–392.
70. Frank, A.; Asuncion, A. *UCI Machine Learning Repository*; University of California, School of Information and Computer Science: Irvine, CA, USA, 2010.
71. Awesome Data. Available online: http://www.awesomedata.com/ (accessed on 16 September 2015).
72. Human Development Reports. Available online: http://hdr.undp.org/en/data# (accessed on 15 September 2019).