



# Kernel Generalized Half-Quadratic Correntropy Conjugate Gradient Algorithm for Online Prediction of Chaotic Time Series

Huijuan Xia<sup>1</sup> · Weijie Ren<sup>2</sup> · Min Han<sup>3,4</sup> 

Received: 12 July 2022 / Revised: 29 November 2022 / Accepted: 30 November 2022 /

Published online: 29 December 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

Kernel adaptive filter armed with information theoretic learning has gained popularity in the domain of time series online prediction. In particular, the generalized correntropy criterion (GCC), as a nonlinear similarity measure, is robust to non-Gaussian noise or outliers in time series. However, due to the nonconvex nature of GCC, optimal parameter estimation may be difficult. Therefore, this paper deliberately combines it with half-quadratic (HQ) optimization to generate the generalized HQ correntropy (GHC) criterion, which provides reliable calculations for convex optimization. After that, a novel adaptive algorithm called kernel generalized half-quadratic correntropy conjugate gradient (KGHCG) algorithm is designed by integrating GHC and the conjugate gradient method. The proposed approach effectively enhances the robustness of non-Gaussian noise and greatly improves the convergence speed and filtering accuracy, and its sparse version KGHCG-VP limits the dimension of the kernel matrix through vector projection, which successfully handles the bottleneck of high computational complexity. In addition, we also discuss the convergence properties, computational

---

✉ Min Han  
minhan@dlut.edu.cn

Huijuan Xia  
xiahuijuan@mail.dlut.edu.cn

Weijie Ren  
renweijie@hrbeu.edu.cn

- <sup>1</sup> Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China
- <sup>2</sup> College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China
- <sup>3</sup> Key Laboratory of Intelligent Control and Optimization for Industrial Equipment of Ministry of Education, Dalian University of Technology, Dalian 116024, China
- <sup>4</sup> Professional Technology Innovation Center of Distributed Control for Industrial Equipment of Liaoning Province, Dalian University of Technology, Dalian 116024, China

complexity and memory requirements in terms of theoretical analysis. Finally, online prediction simulation results with the benchmark Mackey–Glass chaotic time series and real-world datasets show that KGHCG and KGHCG-VP have better convergence and prediction performance.

**Keywords** Kernel adaptive filter · Generalized correntropy criterion · Half-quadratic optimization · Conjugate gradient method

## 1 Introduction

The time series is regarded as a collection of data arranged in chronological order, and it is ubiquitous in nature, industrial production, financial technology, and other fields [11, 16, 17]. Generally, the time series extracted from the practical system has chaotic characteristics, which plays an important role in exploring the evolution rules of the system. However, the real-time property of streaming data and the complexity of the environment bring challenges for the accurate learning of chaotic dynamical systems [25]. Therefore, while mining the hidden information of time series, online prediction models also require to show strong performance for nonlinear, nonstationary, and non-Gaussian aspects of time series. In recent years, kernel adaptive filter (KAF) [21, 26] with universal approximation ability and excellent online learning ability has shown great vitality in various applications, such as time series prediction, channel equalization, nonlinear acoustic echo cancellation, etc. [1, 28, 33].

In the research of KAF, choosing a flexible and robust criterion [13, 14, 29] is crucial. Most classic KAF algorithms including kernel least mean square (KLMS) [20] and kernel recursive least squares (KRLS) [10] are presented under mean square error (MSE) which excel in terms of smoothness, convexity, and computational complexity. However, since MSE contains only second-order statistics and relies on Gaussian noise assumptions, it is difficult to handle the sharp spike and tail heaviness of signal noise in the environment, especially non-Gaussian noise. In order to improve the reliability and robustness of KAF in non-Gaussian environments, the maximum correntropy criterion (MCC) [15, 19, 34] based on information theoretic learning has received extensive attention. Replacing MSE in KRLS with MCC, kernel recursive maximum correntropy (KRMC) [35] is developed to enhance robustness to non-Gaussian noise. To further suppress the interference of nonzero mean noise, KRMC with variable center (KRMC-VC) [22] is proposed. Moreover, as a family criterion of correntropy, the generalized correntropy criterion (GCC) [6, 18] with more nonquadratic loss features is studied, and kernel recursive generalized maximum correntropy (KRGMC) [39] under GCC is developed to improve the flexibility and accuracy of correntropy. However, since GCC is not a strictly convex function [6], the existence of an optimal solution may not be guaranteed. Fortunately, the half-quadratic (HQ) [12, 36] theory can successfully convert a nonconvex problem into a global convex optimization problem, which ensures the best parameter estimates.

In addition, the optimization strategy [8, 9, 27] for obtaining optimal parameter estimates of criteria is also particularly attractive. Stochastic gradient descent (SGD) [23] is the most frequently adapted convex optimization method. However, it is diffi-

cult to guarantee both convergence speed and steady-state performance through SGD optimization. Afterward, optimization strategies based on recursive calculation or second-order optimization [7, 39] are proposed. Although they provide a more accurate solution, they take up excessive calculation and storage space. For the flaws mentioned above, the conjugate gradient (CG) [5] method comes up with a better solution. The nonlinear acoustic echo algorithm [3] achieves a good trade-off between complexity and performance by using the CG method. Kernel conjugate gradient (KCG) [37] algorithm not only solves the problem of slow convergence but also successfully avoids huge computing resources. Last but not least, suitable sparsification methods [2, 31], such as approximate linear dependency (ALD) [10, 26], coherence criterion (CC) [4, 38], and vector projection (VP) [40], are applied in KAF to deal with the high computational cost caused by the rapid growth of the kernel matrix, thereby improving the convergence speed. KCG-AC [37] directly controls data size growth through angle criteria (AC). Quantized KRGMC (QKRGMC) [30] is proposed to quantify the size of the kernel network in the input space by vector quantization.

Based on the above discussion, the main contributions are as follows:

- (1) The generalized HQ correntropy (GHC) criterion is proposed by combining HQ and GCC for the first time. It converts the maximum GCC into a global convex optimization problem via HQ, which makes the parameter estimation more accurate and further enhances the robustness with respect to non-Gaussian noise or outliers.
- (2) Furthermore, the KGHCG algorithm is proposed, which uses the CG method to solve the above GHC criterion of kernel space. The proposed algorithm can provide excellent convergence performance and high filtering accuracy. In addition, the VP method further constrains the infinite expansion mode of the kernel matrix in KGHCG, which greatly reduces the computational complexity, and we finally develop the KGHCG-VP algorithm.
- (3) Finally, we analyze the convergence performance, computational complexity and memory usage of the proposed method. Meanwhile, the efficiency of the proposed algorithms is verified by the Mackey–Glass (MG) dataset as well as real-world datasets including ENSO and Beijing air quality time series. Theoretical analysis and experimental results demonstrate that KGHCG and KGHCG-VP have better prediction ability and practicability for online tasks.

The rest of this article is structured as follows. In Sect. 2, related work is described briefly, including online kernel learning, GCC function, and its nonconvexity. In Sect. 3, we derive the proposed GHC criterion and KGHCG algorithm in detail, and the theoretical analysis of the proposed method is also investigated. In Sect. 4, simulation experiments are given. Finally, we summarize the conclusion in Sect. 5.

## 2 Related Works

### 2.1 Online Kernel Learning

Due to the complexity and dynamic nature of sequential arrival data flow, an online prediction model must demonstrate significant learning capacity for nonlinear and

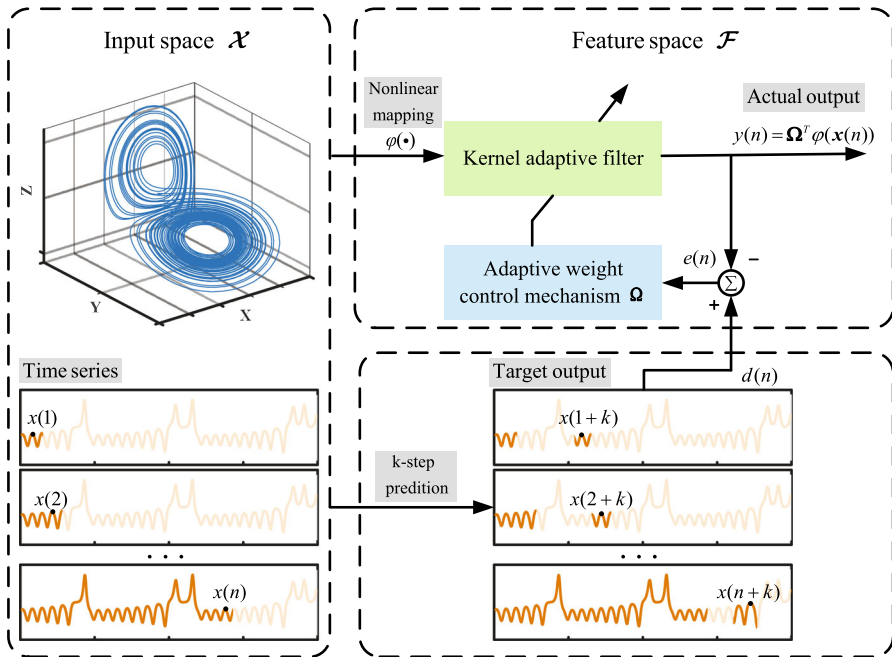


Fig. 1 Kernel adaptive filter for online multi-step prediction

nonstationary. In recent years, KAF has become an efficient nonlinear modeling tool because of its excellent approximation ability and online learning ability. Its core concept is to exploit the online kernel learning framework, and the working principle of KAF is shown in Fig. 1.

For online learning of chaotic time series  $\{x(n), d(n)\}$ , the classical KLMS [20] algorithm transforms the input of the original space  $\mathcal{X}$  into a suitable high-dimensional feature space  $\mathcal{F}$  through a nonlinear mapping  $\varphi(\cdot)$  induced by kernel evaluation. Afterward, a suitable linear method is applied to the transformed sample and the expected output at each iteration  $n$  is estimated as

$$y(n) = \mathbf{\Omega}^T \varphi(x(n)) \tag{1}$$

where  $\mathbf{\Omega}$  is the weight of the filter. According to the adaptive mechanism of KLMS, the learning rule for the weight is

$$\mathbf{\Omega}(n) = \mathbf{\Omega}(n - 1) + \eta e(n) \varphi(x(n)) \tag{2}$$

where  $e(n) = d(n) - y(n)$  denotes the prediction error.  $\eta$  denotes the step parameter.

### 2.2 Generalized Correntropy Criterion

GCC [6] as the measure of the generalized similarity between two random variables  $A$  and  $B$ , is described by

$$L(A, B) = E[\kappa(A, B)] = \iint \kappa(A, B) f_{A,B}(a, b) da db \quad (3)$$

where  $\kappa(\cdot, \cdot)$  denotes the Mercer kernel,  $E(\cdot)$  denotes the mathematical expectation, and  $f_{A,B}(a, b)$  denotes the joint density function of  $A$  and  $B$ . Nevertheless, since  $f_{A,B}(a, b)$  denotes an unknown function in practical applications, the mathematical expectation of the above formula can be approximated empirically through observed samples, which is

$$\hat{L}(A, B) = \frac{1}{N} \sum_{n=1}^N \kappa(a(n), b(n)) \quad (4)$$

where  $\kappa(\cdot, \cdot)$  adopts the generalized Gaussian density function, and it is defined as:

$$\kappa(a, b) = \frac{\alpha}{2\beta\Gamma(1/\alpha)} \exp(-\beta^{-\alpha}|a - b|^\alpha) = \gamma_{\alpha,\beta} \exp(-\lambda|a - b|^\alpha) \quad (5)$$

where  $\gamma_{\alpha,\beta} = \frac{\alpha}{2\beta\Gamma(1/\alpha)}$ ,  $\lambda = \beta^{-\alpha}$ ,  $\alpha > 0$  denotes the shape factor,  $\beta > 0$  denotes the scale factor,  $\Gamma(\cdot)$  denotes the gamma function, and  $\gamma_{\alpha,\beta}$  denotes the normalization constant.

Similar to MCC applied to KAF, the optimal weight vector of the filter can usually be solved by minimizing the generalized correntropy loss (GCL) function [30], and it is

$$\begin{aligned} \ell_{\text{GCL}} &= \kappa_{\alpha,\beta}(0) - E[\kappa_{\alpha,\beta}(e)] = \gamma_{\alpha,\beta} - \frac{1}{N} \sum_{n=1}^N \kappa_{\alpha,\beta}(e(n)) \\ &= \gamma_{\alpha,\beta} \left[ 1 - \frac{1}{N} \sum_{n=1}^N \exp(-\lambda|e(n)|^\alpha) \right] \end{aligned} \quad (6)$$

where  $e(n) = a(n) - b(n)$ . Next, the Hessian matrix with respect to the error  $e$  of  $\ell_{\text{GCL}}$  is calculated to investigate the properties of the GCL function, which is

$$\begin{aligned} \mathbf{H}_{\text{GCL}}(e) &= -\frac{\alpha\lambda\gamma_{\alpha,\beta}}{N} \text{diag}(\Delta_1(\alpha\lambda|e(1)|^\alpha - (\alpha - 1)), \dots, \\ &\quad \Delta_N(\alpha\lambda|e(N)|^\alpha - (\alpha - 1))) \end{aligned} \quad (7)$$

where  $\Delta_n = |e(n)|^{\alpha-2} \exp(-\lambda|e(n)|^\alpha)$ . From obtained Hessian matrix (7), we can obtain two properties as follows:

- (1) if  $0 < \alpha \leq 1$ ,  $\mathbf{H}_{\text{GCL}}(e) \leq 0$  for any  $e$  with  $e(n) \neq 0 (n = 1, \dots, N)$ ;
- (2) if  $\alpha > 1$ ,  $\mathbf{H}_{\text{GCL}}(e) \geq 0$  for any  $e$  with  $|e(n)| \leq [(\alpha - 1)/\alpha\lambda]^{1/\alpha} (n = 1, \dots, N)$ .

From the above properties we can get that if and only if  $\alpha > 1$  and  $|e(n)| \leq [(\alpha - 1)/\alpha\lambda]^{1/\alpha} (n = 1, \dots, N)$ ,  $\mathbf{H}_{\text{GCL}}(e) \geq 0$ . That is, the GCL is convex. When these conditions are not satisfied, GCL as the cost function is not strictly globally convex. As a result, the optimal value cannot be found when solving the cost function.

### 3 Proposed Algorithm

In this section, we derive the KGHCG algorithm in detail. Firstly, we create a GHC function based on GCC function and HQ modeling. Then, the parameters are optimally solved by the CG method for the GHC function of the kernel space. In addition, an online VP approach is adopted to propose KGHCG-VP to reduce the computational complexity. Finally, convergence properties and complexity of proposed method are investigated.

#### 3.1 Generalized Half-Quadratic Correntropy

As mentioned in Sect. 2.2, the Hessian matrix of the GCL function is positive-definite if and only if certain conditions are met. In other words, the global convexity of GCL is difficult to guarantee since its Hessian matrix is not strictly positive definite, and thus GCL cannot directly handle convex optimization tasks. Luckily, the HQ framework ensures that the objective function is strictly convex. Specifically, a GHC criterion is proposed by introducing the intermediate variable  $\mathbf{V}$  to transform nonconvex issues into fully convex issues.

The GCC criteria contain an exponential function as  $f(x) = \exp(-x)$ , and its conjugate function is  $\tilde{f}(v) = v - v \ln(-v)$  with  $v < 0$ . For the convex function of the GCC criteria, the following proposition is made.

**Proposition 1** *The exponential function of GCC is conjugate functions of the convex functions  $\tilde{f}(v) = -v \ln(-v) + v$  with  $v < 0$ . That is*

$$\exp(-\lambda|e(n)|^\alpha) = \sup_{v < 0} \left\{ v\lambda|e(n)|^\alpha - \tilde{f}(v) \right\} \quad (8)$$

where the upper bound value is obtained at  $v = -\exp(-\lambda|e(n)|^\alpha)$ .

**Proof** By the conjugate function theory, we have the conjugate function of  $\tilde{f}(v)$ , which is

$$\tilde{f}^*(u) = \sup_{v < 0} \left\{ uv - \tilde{f}(v) \right\} = \sup_{v < 0} \{ uv - v + v \ln(-v) \} \quad (9)$$

Then, we set  $h(v) = uv - v + v \ln(-v)$ .  $h(v)$  reaches its maximum value when  $v = -\exp(-u)$ , expressed as  $h_{\max}(v) = \exp(-u)$ . Therefore, rewritten formula (9) is given by

$$\tilde{f}^*(u) = \sup_{v < 0} \{ uv - v + v \ln(-v) \} = \exp(-u) \quad (10)$$

where  $v = -\exp(-u)$ . In addition, we make  $u = \lambda|e(n)|^\alpha$  and obtain

$$\tilde{f}^*(\lambda|e(n)|^\alpha) = \sup_{v < 0} \left\{ v\lambda|e(n)|^\alpha - v + v \ln(-v) \right\} = \exp(-\lambda|e(n)|^\alpha) \quad (11)$$

where the upper bound value is obtained when  $v = -\exp(-u)$ , the maximum value is  $\exp(-\lambda|e(n)|^\alpha)$ , and the proposition is proved.  $\square$

Based on the above discussion, the solution of GCL objective function (6) can be equivalent to solving the following optimization function; we have

$$\max_{v < 0} \sum_{n=1}^N \left\{ v(n)\lambda|e(n)|^\alpha - \tilde{f}(v(n)) \right\} \tag{12}$$

For a given  $v(n)$ , optimization objective (12) is equivalent to minimizing the weighted least squares problem by GHC objective function, which is

$$\min_{v < 0} \sum_{n=1}^N \left( -v(n)\lambda|e(n)|^{\alpha-2}e(n)^2 \right) \tag{13}$$

where  $v(n) = -\exp(-\lambda|e(n)|^\alpha)$ .

Next, the Hessian matrix of GHC function (13) is calculated, and we get

$$\mathbf{H}_{\text{GHC}}(e) = \text{diag} \left( -2\lambda v(1)|e(1)|^{\alpha-2}, -2\lambda v(2)|e(2)|^{\alpha-2}, \dots, -2\lambda v(N)|e(N)|^{\alpha-2} \right) \tag{14}$$

Since  $v(n) < 0$ , we can obtain Hessian matrix (14) of the GHC loss function strictly positive definite, ensuring that GHC is a strictly convex function. Compared to GCC, objective function (13) solves nonconvex to global convex optimization. Then, we utilize the conjugate gradient method to calculate.

First, we define minimizing GHC function (13) as a quadratic function optimization objective, which is given by

$$\min \frac{1}{2} \left\| \sqrt{\mathbf{V}} \left( \mathbf{d}^T - \mathbf{U}^T \mathbf{w} \right) \right\|^2 \Rightarrow \min_{\mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{U} \mathbf{V} \mathbf{U}^T \mathbf{w} - \mathbf{d} \mathbf{V} \mathbf{U}^T \mathbf{w} \tag{15}$$

where  $\mathbf{d} = [d(1), d(2), \dots, d(N)]$ ,  $\mathbf{U} = [\mathbf{u}(1), \mathbf{u}(2), \dots, \mathbf{u}(N)]$ ,  $\mathbf{w}$  is the weight vector, and  $\mathbf{V}$  is the intermediate variable, which is denoted as

$$\mathbf{V} = \text{diag}(-2v(1)\lambda|e(1)|^{\alpha-2}, \dots, -2v(N)\lambda|e(N)|^{\alpha-2}) \tag{16}$$

Then, we utilize the CG approach to minimize the GHC objective function to propose the GHC-CG algorithm, which is similar to CGLS [24]. For a given data stream  $\{\mathbf{u}(n), d(n)\}$ , initializations are set to  $\mathbf{w}(0) = 0$ ,  $\mathbf{r}(0) = \mathbf{V}(\mathbf{d}^T - \mathbf{U}^T \mathbf{w}(0))$ ,

$s(0) = \mathbf{U}\mathbf{r}(0)$  and  $\mathbf{p}(1) = s(0)$ , and the optimization mechanisms are described as:

$$\begin{cases} \mathbf{v}(n) = \mathbf{V}\mathbf{U}^T \mathbf{p}(n) \\ \varsigma_1(n) = \langle s(n-1), s(n-1) \rangle / \langle \mathbf{U}^T \mathbf{p}(n), \mathbf{v}(n) \rangle \\ \mathbf{w}(n) = \mathbf{w}(n-1) + \varsigma_1(n) \mathbf{p}(n) \\ \mathbf{r}(n) = \mathbf{r}(n-1) - \varsigma_1(n) \mathbf{v}(n) \\ s(n) = \mathbf{U}\mathbf{r}(n) \\ \varsigma_2(n+1) = \langle s(n), s(n) \rangle / \langle s(n-1), s(n-1) \rangle \\ \mathbf{p}(n+1) = s(n) + \varsigma_2(n+1) \mathbf{p}(n) \end{cases} \quad (17)$$

where  $\mathbf{v}$  denotes the intermediate vector,  $\varsigma_1$  and  $\varsigma_2$  denote learning parameters,  $\mathbf{r}$  is the residual vector,  $s$  denotes the residual vector of normal equations, and  $\mathbf{p}$  denotes the search direction.

### 3.2 Kernel Generalized Half-Quadratic Correntropy Conjugate Gradient Algorithm

According to the GHC objective function, the weighted least squares problem is applied to the kernel space, which is

$$\min_{v < 0} \sum_{n=1}^N \left( -v(n) \lambda |e(n)|^{\alpha-2} e(n)^2 \right) = \min_{\theta} \frac{1}{2} \left\| \sqrt{\mathbf{V}} \left( \mathbf{d}^T - \mathbf{K}\theta \right) \right\|^2 \quad (18)$$

where  $\theta$  is the expansion coefficient, which needs to be estimated.  $\mathbf{K}$  is the kernel matrix, which is expressed as

$$\mathbf{K} = \begin{bmatrix} \kappa(\mathbf{u}(1), \mathbf{u}(1)) & \kappa(\mathbf{u}(1), \mathbf{u}(2)) & \cdots & \kappa(\mathbf{u}(1), \mathbf{u}(N)) \\ \kappa(\mathbf{u}(2), \mathbf{u}(1)) & \kappa(\mathbf{u}(2), \mathbf{u}(2)) & \cdots & \kappa(\mathbf{u}(2), \mathbf{u}(N)) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\mathbf{u}(N), \mathbf{u}(1)) & \kappa(\mathbf{u}(N), \mathbf{u}(2)) & \cdots & \kappa(\mathbf{u}(N), \mathbf{u}(N)) \end{bmatrix}_{N \times N} \quad (19)$$

Then, we minimize the GHC function using the CG method. For the KGHCG algorithm, the expansion coefficient  $\theta$  and kernel matrix  $\mathbf{K}$  update mechanism are very important. We define the kernel matrix  $\mathbf{K}$  as

$$\mathbf{K}(N) = \begin{bmatrix} \mathbf{K}(N-1) & \mathbf{g}(N)^T \\ \bar{\mathbf{g}}(N) & q(N) \end{bmatrix}_{N \times N} \quad (20)$$

where  $\mathbf{g}(N) = [\kappa(\mathbf{u}(1), \mathbf{u}(1)), \dots, \kappa(\mathbf{u}(N-1), \mathbf{u}(N))]$  and  $q(N) = \kappa(\mathbf{u}(N), \mathbf{u}(N))$ .



The weight vector  $\mathbf{w}(N)$  is obtained by kernel trick and mathematical induction.

$$\begin{aligned}
 \mathbf{w}(N) &= \mathbf{w}(0) + \sum_{n=1}^N \varsigma_1(n) \mathbf{p}(n) \\
 &= \sum_{n=1}^N \varsigma_1(n) \sum_{i=1}^N \pi_{i+1}^i s(i-1) \\
 &= \sum_{n=1}^N \left( \sum_{i=n}^N \varsigma_1(i) \pi_{n+1}^i \right) \mathbf{s}(n-1) \\
 &= \mathbf{U} \left[ \sum_{n=1}^N \left( \sum_{i=n}^N \varsigma_1(i) \pi_{n+1}^i \right) \mathbf{r}(n-1) \right] \\
 &= \mathbf{U} (\mathbf{E}(N) \boldsymbol{\xi}(N)) = \mathbf{U} \boldsymbol{\theta}(N)
 \end{aligned} \tag{21}$$

where  $\pi_{n+1}^i = \varsigma_{2n+1} \varsigma_{2n+2} \cdots \varsigma_{2i}$  with  $\pi_{n+1}^n = 1$ .  $\mathbf{E}(N) = [\mathbf{r}(0), \mathbf{r}(1), \dots, \mathbf{r}(N-1)]$ ,  $\boldsymbol{\theta}(N) = \mathbf{E}(N) \boldsymbol{\xi}(N)$ , where the  $n$ th element of  $\boldsymbol{\xi}(N)$  is defined by

$$\xi_n(N) = \sum_{i=n}^N \varsigma_1(i) \pi_{n+1}^i = \xi_n(N-1) + \varsigma_1(N) \pi_{n+1}^N \tag{22}$$

Since  $[\boldsymbol{\theta}(N-1); 0]$  is a good approximation of  $\boldsymbol{\theta}(N)$ , it does not need to iterate over the number of data streams, and only needs one or two iterations to obtain superior parameter performance. Hence,  $[\boldsymbol{\theta}(N-1); 0]$  is used as the initial value to replace  $\boldsymbol{\theta}(N)$ . And the update of  $\boldsymbol{\theta}(N)$  about the KGHCG algorithm with two iterations is expressed as

$$\boldsymbol{\theta}(N) = [\boldsymbol{\theta}(N-1); 0] + (\varsigma_1(1) + \varsigma_2(2) \varsigma_1(2)) \mathbf{r}(0) + \varsigma_1(2) \mathbf{r}(1) \tag{23}$$

At this point, the initial residual calculation formula is given by

$$\begin{aligned}
 \mathbf{r}(0) &= \mathbf{V}(N) (\mathbf{d}^T - \mathbf{K}(N) [\boldsymbol{\theta}(N-1); 0]) \\
 &= \mathbf{V}(N) \left( \begin{bmatrix} \mathbf{d}(N-1)^T \\ d(N) \end{bmatrix} - \begin{bmatrix} \mathbf{K}(N-1) & \mathbf{g}(N)^T \\ \bar{\mathbf{g}}(N) & q(N) \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}(N-1) \\ 0 \end{bmatrix} \right) \\
 &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0}^T & v(N) \end{bmatrix} \begin{bmatrix} (\mathbf{d}(N-1) - \boldsymbol{\theta}(N-1)^T \mathbf{K}(N-1))^T \\ d(N) - \bar{\mathbf{g}}(N) \boldsymbol{\theta}(N-1) \end{bmatrix} \\
 &= [\mathbf{e}(N-1), v(N) (d(N) - \bar{\mathbf{g}}(N) \boldsymbol{\theta}(N-1))]^T
 \end{aligned} \tag{24}$$

where  $v(N) = 2\lambda |e(N)|^{\alpha-2} \exp(-\lambda |e(N)|^\alpha)$ ,  $\mathbf{e}(N) = (\mathbf{r}(1) - \varsigma_1(2) \mathbf{v}(2))^T$  and  $e(N) = (d(N) - \bar{\mathbf{g}}(N) \boldsymbol{\theta}(N-1))$ .

In addition, since the size of the kernel matrix is determined by data scale, more and more computing resources need to be undertaken with the passage of time. To deal with the drawback of infinite expansion of the kernel matrix, the VP method [41] as the sparsification strategy is used. This criterion generates a more compact network by selecting samples with valid information, which further reduces the computational

complexity. And it is defined as

$$\cos(\mathbf{u}, \mathbf{u}') = \frac{\langle \varphi(\mathbf{u}), \varphi(\mathbf{u}') \rangle_{\mathcal{F}}}{\|\varphi(\mathbf{u})\|_{\mathcal{F}} \|\varphi(\mathbf{u}')\|_{\mathcal{F}}} \tag{25}$$

Then we compare the distance  $\cos(\mathbf{u}, \mathbf{u}')$  with the predefined threshold  $\tau$  that determines the level of sample sparseness. If  $\cos(\mathbf{u}, \mathbf{u}') > \tau$ ,  $(\varphi(\mathbf{u}), d(n))$  will be discarded. Otherwise,  $(\varphi(\mathbf{u}), d(n))$  will be added to the existing dictionary  $\mathcal{D}$ , and the expansion coefficient and residual will be updated in real time according to the adaptive mechanism, achieving a good compromise between prediction accuracy and computational efficiency.

The procedure of the KGHCG-VP algorithm is detailed in Algorithm 1.

---

**Algorithm 1:** online KGHCG-VP algorithm

---

**Input:** samples  $\{\mathbf{u}(n), d(n)\}$

**Output:**  $\hat{d}(n + 1)$

**Parameters setting:** Gaussian kernel width  $\sigma$ , shape parameter  $\alpha$ , scale parameter  $\beta$ , dictionary size  $D$

**Initialization:**  $\mathbf{U}(1) = \mathbf{u}(1)$ ,  $q(1) = \kappa(\mathbf{u}(1), \mathbf{u}(1))$ ,  $\mathbf{q}(1) = \sqrt{q(1)}$ ,  $\mathbf{K}(1) = q(1)$ ,  $\theta(1) = \hat{d}(1)/q(1)$ ,  $\mathbf{e}(1) = 0$ ,  $D = 1$

```

while  $n = 2, 3, \dots, N$  do
     $\{\mathbf{u}(n), d(n)\}$  is available
     $q(n) = \kappa(\mathbf{u}(n), \mathbf{u}(n))$ 
     $\mathbf{g}(n) = [\kappa(\mathbf{u}_D(:, 1), \mathbf{u}(n)), \dots, \kappa(\mathbf{u}_D(:, D), \mathbf{u}(n))]$ 
     $\cos(\mathbf{u}(n), \mathbf{u}_D) = \mathbf{g}_n(D) / [\sqrt{q(n)}q_{n-1}(D)]$ 
    if  $\max\{|\cos(\mathbf{u}(n), \mathbf{u}_D)|\} < \tau$  then
         $D = D + 1$ 
         $\mathbf{U}(D) = [\mathbf{U}(D - 1), \mathbf{u}(n)]$ ,  $\mathbf{q}(D) = [q(D - 1), \sqrt{q(n)}]$ 
         $\mathbf{K}(D) = \begin{bmatrix} \mathbf{K}(D - 1) & \mathbf{g}(n)^T \\ \tilde{\mathbf{g}}(n) & q(n) \end{bmatrix}$ 
         $\mathbf{e}(n) = d(n) - \tilde{\mathbf{g}}(n)\theta(D - 1)$ 
         $\mathbf{v}(n) = 2\beta^{-\alpha} |e(n)|^{\alpha-2} \exp(-\beta^{-\alpha} |e(n)|^\alpha)$ 
         $\mathbf{r}(0) = [\mathbf{e}(D - 1), \mathbf{v}(n)\mathbf{e}(n)]^T$ ,  $\mathbf{v}(1) = \mathbf{K}(D)\mathbf{r}(0)$ 
         $\mu(0) = \langle \mathbf{v}(1), \mathbf{r}(0) \rangle$ ,  $\varsigma_1(1) = \mu(0) / \langle \mathbf{v}(1), \mathbf{v}(1) \rangle$ 
         $\mathbf{r}(1) = \mathbf{r}(0) - \varsigma_1 \mathbf{v}(1)$ ,  $\mathbf{s}(1) = \mathbf{K}(D)\mathbf{r}(1)$ 
         $\mu(1) = \langle \mathbf{s}(1), \mathbf{r}(1) \rangle$ ,  $\varsigma_2(2) = \mu(1) / \mu(0)$ 
         $\mathbf{v}(2) = \varsigma_2(2)\mathbf{v}(1) + \mathbf{s}(1)$ ,  $\varsigma_1(2) = \mu(1) / \langle \mathbf{v}(2), \mathbf{v}(2) \rangle$ 
         $\theta(D) = [\theta(D - 1); 0] + (\varsigma_1(1) + \varsigma_2(2)\varsigma_1(2))\mathbf{r}(0) + \varsigma_1(2)\mathbf{r}(1)$ 
         $\mathbf{e}(D) = (\mathbf{r}(1) - \varsigma_1(2)\mathbf{v}(2))^T$ 
    end
end

```

---

### 3.3 Convergence Analysis of KGHCG

For the proposed KGHCG algorithm, the update rules of weight and direction vectors of (17) are as follows

$$\begin{cases} \mathbf{w}(n+1) = \mathbf{w}(n) + \zeta_1(n) \mathbf{p}(n) \\ \mathbf{p}(n+1) = \mathbf{s}(n) + \zeta_2(n) \mathbf{p}(n) \end{cases} \quad (26)$$

where  $\mathbf{s}(n)$  is the residual vector of normal equations (negative gradient vector), and the core concept of the direction vector is to search for the minimum value of the loss function, where  $\lim_{n \rightarrow \infty} \inf \|\mathbf{s}(n)\| = 0$ .

In [42], descent conditions  $\mathbf{s}(n) \mathbf{p}(n+1) < 0$  [37] of conjugate gradient and (26) together constitute the Zoutendijk criteria, which are defined as

$$\begin{cases} \mathbf{w}(n+1) = \mathbf{w}(n) + \zeta_1(n) \mathbf{p}(n) \\ \mathbf{s}(n) \mathbf{p}(n+1) < 0 \end{cases} \quad (27)$$

where direction vector  $\mathbf{p}(n)$  and negative gradient vector  $\mathbf{s}(n)$  satisfy the following inequality

$$\sum_{n=1}^{\infty} \frac{(\mathbf{s}(n-1) \mathbf{p}(n))^2}{\|\mathbf{p}(n)\|^2} < +\infty \quad (28)$$

Then, we prove convergence by contradiction.

**Proposition 2** Based on (26), the proposed algorithm converges to the minimum value of the loss function, where

$$\lim_{n \rightarrow \infty} \inf \|\mathbf{s}(n)\| = 0 \quad (29)$$

**Proof** First, we assume that the above Proposition 2 is false, i.e.,  $\lim_{n \rightarrow \infty} \inf \|\mathbf{s}(n)\| \geq \delta$ , where  $\delta$  is an arbitrarily small constant. Multiplying both sides of the negative direction vector  $\mathbf{s}(n)$  in (26), we have

$$\begin{aligned} \mathbf{s}(n-1)^T \mathbf{p}(n+1) &= \mathbf{s}(n-1)^T \mathbf{s}(n) + \mathbf{s}(n-1)^T \zeta_2(n+1) \mathbf{p}(n) \\ \Rightarrow \zeta_2(n+1) &= \frac{\mathbf{s}(n-1)^T \mathbf{p}(n+1)}{\mathbf{s}(n-1)^T \mathbf{p}(n)} \approx \frac{\mathbf{s}(n)^T \mathbf{p}(n+1)}{\mathbf{s}(n-1)^T \mathbf{p}(n)} \end{aligned} \quad (30)$$

□

Then, calculating the 2-norm on both sides of the direction vector  $\mathbf{p}(n+1)$  of (26) at the same time, and it is

$$\begin{aligned} \|\mathbf{p}(n+1)\|^2 &= \|\mathbf{s}(n) + \zeta_2(n+1) \mathbf{p}(n)\|^2 \\ &= \|\mathbf{s}(n)\|^2 + 2\mathbf{s}(n)^T \zeta_2(n+1) \mathbf{p}(n) + \zeta_2(n+1)^2 \|\mathbf{p}(n)\|^2 \\ &= \|\mathbf{s}(n)\|^2 + 2\mathbf{s}(n)^T [\mathbf{p}(n+1) - \mathbf{s}(n)] + \zeta_2(n+1)^2 \|\mathbf{p}(n)\|^2 \\ &= \zeta_2(n+1)^2 \|\mathbf{p}(n)\|^2 + 2\mathbf{s}(n)^T \mathbf{p}(n+1) - \|\mathbf{s}(n)\|^2 \end{aligned} \quad (31)$$

Substituting the learning factor into formula (31), we obtain

$$\|p(n + 1)\|^2 = \left( \frac{s(n)^T p(n + 1)}{s(n - 1)^T p(n)} \right)^2 \|p(n)\|^2 + 2s(n)^T p(n + 1) - \|s(n)\|^2 \tag{32}$$

Next, dividing both sides of Eq. (32) by  $(s(n)^T p(n + 1))^2$ , we have

$$\begin{aligned} \frac{\|p(n + 1)\|^2}{(s(n)^T p(n + 1))^2} &= \frac{\|p(n)\|^2}{(s(n)^T p(n + 1))^2} + \frac{2}{s(n)^T p(n + 1)} - \frac{\|s(n)\|^2}{(s(n)^T p(n + 1))^2} \\ &= \frac{\|p(n)\|^2}{(s(n)^T p(n + 1))^2} - \left( \frac{\|s(n)\|}{s(n)^T p(n + 1)} - \frac{1}{\|s(n)\|} \right) + \frac{1}{\|s(n)\|^2} \\ &\leq \frac{\|p(n)\|^2}{(s(n)^T p(n + 1))^2} + \frac{1}{\|s(n)\|^2} \end{aligned} \tag{33}$$

Further, we can obtain the scaling inequality by

$$\begin{aligned} \frac{(s(n)^T p(n + 1))^2}{\|p(n)\|^2} &\geq \left[ \frac{1}{\|s(0)\|^2} + \frac{1}{\|s(1)\|^2} + \dots + \frac{1}{\|s(n - 1)\|^2} \right]^{-1} \\ &= \left( \sum_{i=0}^{n-1} \frac{1}{\|s(i)\|^2} \right)^{-1} \geq \frac{\delta^2}{n} \end{aligned} \tag{34}$$

As  $\sum_{n=1}^{\infty} \delta^2/n = +\infty$ , we get the inequality of direction vector  $p(n)$  and negative gradient vector  $s(n)$

$$\sum_{n=1}^{\infty} \frac{(s(n)^T p(n + 1))^2}{\|p(n)\|^2} \geq +\infty \tag{35}$$

which is different from (28) in the Zoutendijk criteria. So the assumption  $\liminf_{n \rightarrow \infty} \|s(n)\| \geq \delta$  is wrong.

Since  $\lim_{n \rightarrow \infty} \inf \|s(n)\| = 0$  is guaranteed, the convergence of the proposed algorithm can be obtained, and its convergence is further verified by simulation experiments.

### 3.4 Computational Complexity and Memory Usage of KGHCG

According to the complete idea of the proposed algorithm KGHCG, Table 1 shows its computational complexity comparison with KLMS [20], KRMC [35], KRMC-VC [22], and KRGMC [39] at the  $N$ th iteration. Due to its simple structure, although KLMS [20] has the smallest computational complexity, the prediction accuracy and

**Table 1** Computational complexity of different models

Models	Additions	Multiplications	Divisions
KLMS [20]	$N$	$N$	0
KRMC [35]	$4N^2 + 4N$	$4N^2 + 4N + 2$	1
KRMC-VC [22]	$4N^2 + 4N + L + 2$	$4N^2 + 4N + L + 2$	2
KRGMC [39]	$5N^2 + 5N$	$5N^2 + 5N + 4$	4
KGHCG	$2N^2 + 8N$	$2N^2 + 10N + 2$	3

convergence speed are the worst. Because of the introduction of variable centers  $c$ , the number of additions, multiplications, and divisions of KRMC-VC [22] is  $L+2$  ( $L$  is the number of sliding data),  $L$ , and 1 more than that of KRMC [35], respectively. Although the division operation increases at each iteration compared with KRMC [35] and KRMC-VC [22], the coefficient of  $O(N^2)$  of KGHCG is smaller, and computational complexity of KGHCG is still lower than their cost. In comparison with KRGMC [39] and KGHCG based on GCC and its variation GHC, the recursive calculation of the former consumes the largest computational burden, and the latter KGHCG based on the CG method has excellent performance at a low computational burden. It has  $2N^2 + 8N$  additions,  $2N^2 + 10N + 2$  multiplications, and 3 divisions.

Overall, the computational complexity of the proposed KGHCG is in the middle of other algorithms, and its convergence speed and prediction capability are comparable to those of algorithms based on recursive calculation. When all algorithms are sparse, their computational complexity uses the number of dictionaries  $D$  instead of  $N$ , with no extra cost for additions, multiplications, and divisions.

Furthermore, memory usage depends on the size of the kernel matrix of the online algorithms. Specifically, KLMS [20] requires  $O(N)$  memory. The order of memory required for KRMC [35], KRMC-VC [22], and KRGMC [39], and KGHCG is  $O(N^2)$ . When the dictionary size  $D \ll N$ , KGHCG-VP and other sparse algorithms need less memory budget, and their order is  $O(D^2)$ .

## 4 Experimental Results

In this section, the robustness of KGHCG algorithm and its sparse version KGHCG-VP will be verified in three time series. One of them is the benchmark MG chaotic time series with *alpha-stable*, and the others are real-world datasets with ENSO time series and Beijing air quality time series. In addition, a variety of online algorithms, including KLMS [20], KRMC [35], KRGMC [39], KRMC-VC [22], KRMC-ALD, KCG-AC [37], and QKRGMC [30], are used to compare with proposed online algorithms to evaluate the superior performance.

For the online prediction algorithm, its goal is to effectively improve the convergence speed while ensuring the accuracy of time series prediction. As a result, the complexity of the algorithm is characterized by dictionary size, training, and testing time. MSE, root MSE (RMSE), symmetric mean absolute percentage error (SMAPE), and  $R^2$  are used to describe the prediction accuracy, which are given by

$$\left\{ \begin{array}{l} \text{MSE} = \frac{1}{N} \sum_{n=1}^N (y(n) - d(n))^2 \\ \text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^N (y(n) - d(n))^2} \\ \text{SMAPE} = \frac{100\%}{N} \sum_{n=1}^N \frac{2|y(n)-d(n)|}{|y(n)|+|d(n)|} \\ R^2 = 1 - \frac{\sum_{n=1}^N (y(n)-d(n))^2}{\sum_{n=1}^N (d(n)-\bar{d})^2} \end{array} \right. \quad (36)$$

where  $d(n)$  and  $y(n)$  stand for the actual value and the estimated value,  $\bar{d}$  stands for the average of the actual value, and  $N$  stands for the size of the samples.

#### 4.1 Mackey–Glass Time Series

In this part, the benchmark MG chaotic time series is considered to investigate the influence of non-Gaussian noise for the proposed methods, and it is calculated by the nonlinear differential equation:

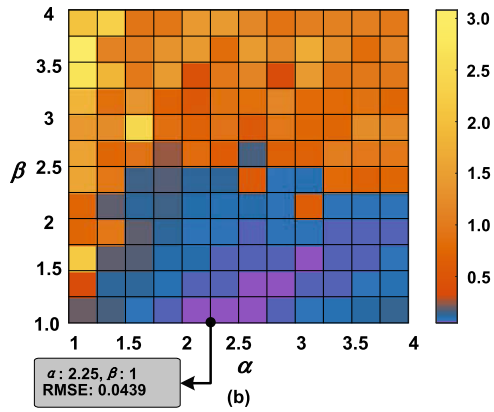
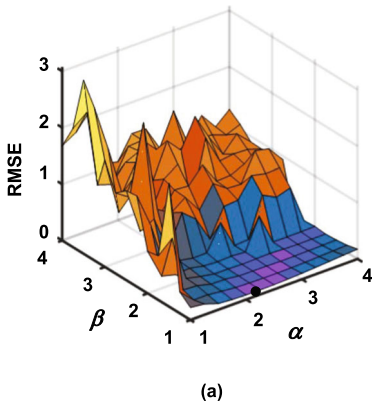
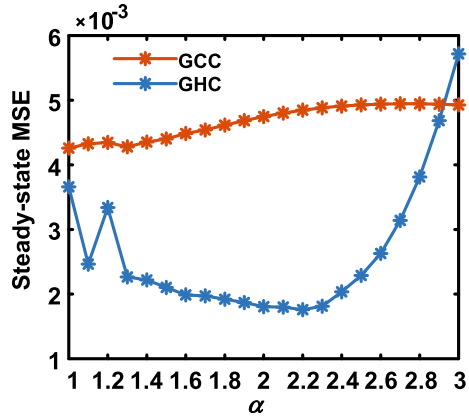
$$\frac{dx(t)}{dt} = 0.9x(t) + \frac{0.2x(t - \zeta)}{1 + x(t - \zeta)^{10}} \quad (37)$$

where  $\zeta = 17$  and the initial value is 0.5. The anti-noise ability of KGHCG algorithm is demonstrated by adding *alpha-stable* noise (the characteristic exponent is 1.8, the skewness is 0.5, the scale and location are 0.0001 and 0.2). In the online task, we make the past 10 steps  $x(k), x(k - 1), \dots, x(k - 9)$  to predict the next 5 step  $x(k + 5)$ . And the shapes of training samples and clean testing samples are  $1500 \times 10$  and  $500 \times 1$ , respectively.

In the first trial, we explore the influence of the shape parameter  $\alpha$  on GCC and the proposed GHC criteria with non-Gaussian noise, and we set both the Gaussian kernel width  $\sigma$  and the scale parameter  $\beta$  to 1. Then, we plot the variation of the steady-state MSE with  $\alpha \in [1, 3]$ . As shown in Fig. 2, the filtering accuracy of the proposed GHC-based KGHCG algorithm dramatically outperforms the competitor in most conditions of  $\alpha$ . In particular, the steady-state MSE of the proposed KGHCG is the smallest at  $\alpha = 2.25$ .

Furthermore, we explore the effects of the shape parameter  $\alpha$  and the scale parameter  $\beta$  of KGHCG. Considering  $\alpha, \beta \in [1, 4]$ , the grid search method is utilized to find the optimal parameter combination. We take the last 300 sets of data from the training set as the validation set and visualize the RMSE index to evaluate the prediction results of KGHCG in Fig. 3. It clearly shows that there is a positive correlation between parameter  $\beta$  and the RMSE index, and the value of RMSE increases after slightly decreases for parameter  $\alpha$  alone. Finally, when the optimal  $\alpha$  and  $\beta$  values are 2.25 and 1, the RMSE value of KGHCG reaches a minimum of 0.0439.

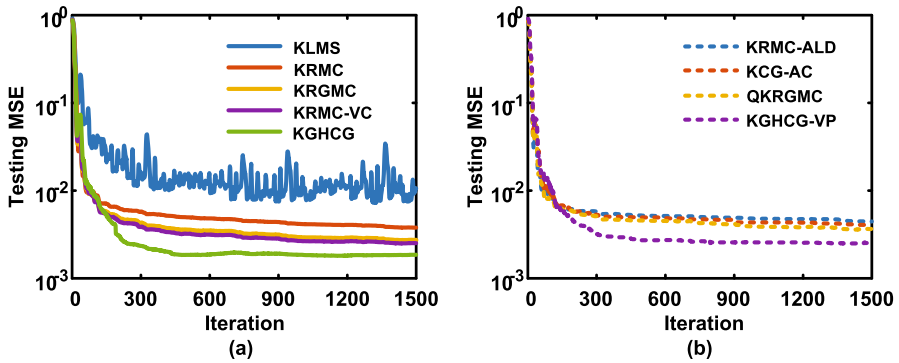
**Fig. 2** The influence of the shape parameter  $\alpha$  under GCC and proposed GHC for MG chaotic time series with  $\alpha$ -stable noise



**Fig. 3** Error situations under different  $\alpha$  and  $\beta$  by KGHCG for MG chaotic time series with  $\alpha$ -stable noise

**Table 2** Parameters setting of different methods for MG chaotic time series ( $\eta$  is the step parameter.  $\gamma$  is the regularization factor.  $\sigma'$  is the bandwidth of the correntropy.  $\alpha$  is the shape factor.  $\beta$  is the scale factor.  $c$  is the center location of the kernel.  $L$  is the number of sliding data.  $\nu$ ,  $\xi$  and  $\tau$  are the predefined threshold.  $\varepsilon$  is the quantization size)

Methods	$\eta$	$\gamma$	$\sigma'$	$\alpha$	$\beta$	$c$	$L$	$\nu$	$\xi$	$\varepsilon$	$\tau$	Kernel type	Kernel width
KLMS [20]	0.2	–	–	–	–	–	–	–	–	–	–	Gauss	1
KRMC [35]	–	0.5	1	–	–	–	–	–	–	–	–	Gauss	1
KRGMC [39]	–	0.5	–	2.25	1	–	–	–	–	–	–	Gauss	1
KRMC-VC [22]	–	0.5	1	–	–	0	1	–	–	–	–	Gauss	1
KGHCG	–	–	–	2.25	1	–	–	–	–	–	–	Gauss	1
KRMC-ALD	–	0.5	1	–	–	–	–	0.017	–	–	–	Gauss	1
KCG-AC [37]	–	–	–	–	–	–	–	–	0.99	–	–	Gauss	1
QKRGMC [30]	–	0.5	–	2.25	1	–	–	–	–	0.115	–	Gauss	1
KGHCG-VP	–	–	–	2.25	1	–	–	–	–	–	0.99	Gauss	1



**Fig. 4** **a** Average learning curves of basic algorithms. **b** Average learning curves of sparse algorithms for MG chaotic time series with  $\alpha$ -stable noise

In the second trial, the robustness of KGHCG and KGHCG-VP is investigated by comparing the two groups, namely basic algorithms and sparse algorithms. Table 2 provides parameters setting of all online algorithms. As shown in Fig. 4, the convergence performance of different algorithms is visualized, and simulation results of different methods are obtained in Table 3. Based on the above experimental results, one can get the following conclusions:

- (1) Due to the poor adaptability of MSE to  $\alpha$ -stable noise, KLMS [20] has the weakest convergence performance. In addition, compared with KRMC [35], KRGMC [39], and KRMC-VC [22], the proposed KGHCG, which combines GHC criterion and CG method, has the strongest adaptability to non-Gaussian noise in Fig. 4a.
- (2) Combining Fig. 4b and Table 3, KGHCG-VP, which introduced the sparse technique VP, can stably converge to the minimum value compared with the other three sparse algorithms under the condition that the dictionary is approximately the same. RMSE, SMAPE, and  $R^2$  of KGHCG-VP are 0.0502, 0.0427, and 0.9507, respectively, following KGHCG and still superior to KLMS [20], KRMC [35], KRGMC [39], and KRMC-VC [22]. Although the accuracy of KGHCG-VP is slightly lower than that of the original KGHCG, computation time is significantly shortened and the predicting speed is accelerated.
- (3) By observing prediction curves and error distributions for the MG chaotic time series in Fig. 5, KGHCG-VP can effectively track the change of the MG chaotic time series with excellent fitting results, and its final error distributions present the characteristics of a normal distribution. From the perspective of sparsity, KGHCG-VP can generate a compact dictionary structure and still maintain good prediction performance.

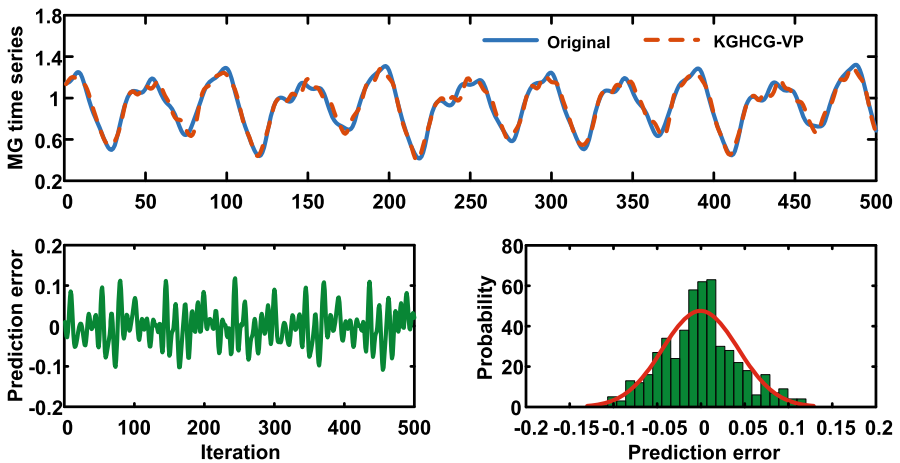
## 4.2 El Nino-Southern Oscillation Time Series

The ENSO dataset is one of the most significant and widely influential chronological climate signals in the Earth's system and has been proven to contain potentially chaotic



**Table 3** Simulation results of different methods for MG chaotic time series

Methods		Training time	Testing time	RMSE	SMAPE	$R^2$	Dictionary
Baseline	KLMS [20]	4.4018	0.0198	0.1029	0.0934	0.7940	1500
	KRMC [35]	11.7934	0.0189	0.0616	0.0548	0.9262	1500
	KRGMC [39]	17.2894	0.0193	0.0522	0.0457	0.9469	1500
	KRMC-VC [22]	11.6462	0.0185	0.0503	0.0435	0.9510	1500
	KGHCG	9.5413	0.0176	0.0431	0.0372	0.9638	1500
Sparsity	KRMC-ALD	0.9400	0.0073	0.0677	0.0594	0.9134	288
	KCG-AC [37]	0.9000	0.0093	0.0638	0.0579	0.9208	276
	QKRGMC [30]	0.7824	0.0096	0.0608	0.0538	0.9287	273
	KGHCG-VP	0.6792	0.0073	0.0502	0.0427	0.9507	282

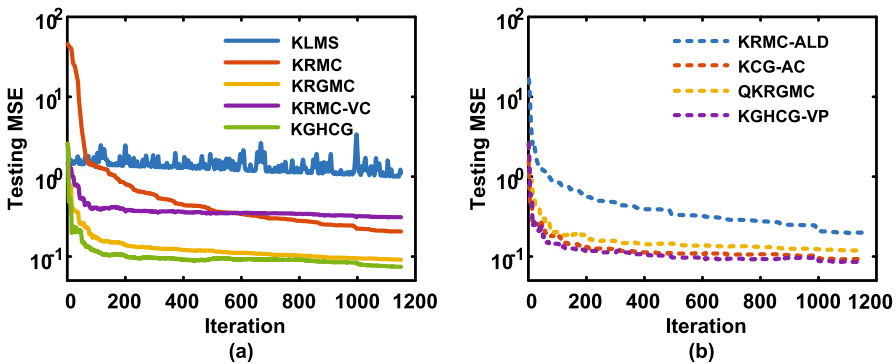
**Fig. 5** Prediction curves and error distributions of KGHCG-VP for MG chaotic time series with  $\alpha$ -stable noise

properties [32]. Therefore, the accurate study of it not only contributes to the early warning of meteorological disasters, but also provides important assistance for the prediction of future climate trends. Hence, climate and sea surface temperature indicators are utilized to make a five-step prediction of the Nino 3.4 indicator. According to the phase space reconstruction theory (PSRT), we use the C-C method to reconstruct 1452 groups of ENSO datasets including monthly Pacific Decadal Oscillation and Southern Oscillation Index, Nino 1.2, Nino 3, Nino 3.4, and Nino 4 from January 1900 to December 2020 from the National Oceanic and Atmospheric Administration (<http://www.psl.noaa.gov/gcosp/Timeseries/>), and calculate that the embedding dimension and the delay time are [2, 2, 2, 2, 3, 3] and [4, 4, 3, 3, 4, 3], respectively. After that, we split the reconstructed data into training samples and testing samples with a ratio of 4:1.

For ENSO time series prediction, the parameters setting of different methods is shown in Table 4. Real-world datasets are often polluted by various adverse factors,

**Table 4** Parameters setting of different methods for ENSO time series ( $\eta$  is the step parameter.  $\gamma$  is the regularization factor.  $\sigma'$  is the bandwidth of the correntropy.  $\alpha$  is the shape factor.  $\beta$  is the scale factor.  $c$  is the center location of the kernel.  $L$  is the number of sliding data.  $\nu$ ,  $\xi$  and  $\tau$  are the predefined threshold.  $\varepsilon$  is the quantization size)

Methods	$\eta$	$\gamma$	$\sigma'$	$\alpha$	$\beta$	$c$	$L$	$\nu$	$\xi$	$\varepsilon$	$\tau$	Kernel type	Kernel width
KLMS [20]	0.1	–	–	–	–	–	–	–	–	–	–	Gauss	47
KRMC [35]	–	0.1	2	–	–	–	–	–	–	–	–	Gauss	47
KRGMG [39]	–	0.1	–	2.5	1	–	–	–	–	–	–	Gauss	47
KRMC-VC [22]	–	0.1	2	–	–	0.5	1	–	–	–	–	Gauss	47
KGHCG	–	–	–	2.5	1	–	–	–	–	–	–	Gauss	47
KRMC-ALD	–	0.1	2	–	–	–	–	0.009	–	–	–	Gauss	47
KCG-AC [37]	–	–	–	–	–	–	–	–	0.995	–	–	Gauss	47
QKRGMC [30]	–	0.1	–	2.5	1	–	–	–	–	2.9	–	Gauss	47
KGHCG-VP	–	–	–	2.5	1	–	–	–	–	–	0.998	Gauss	47



**Fig. 6** **a** Average learning curves of basic algorithms. **b** Average learning curves of sparse algorithms for ENSO time series

which can degrade the performance of prediction models. Therefore, we demonstrate the adaptability and scalability of the proposed algorithms in natural environments by visualizing the testing MSE of different online algorithms for the prediction of Nino 3.4.

- (1) As shown in Fig. 6, whether the online algorithms are sparse or not, the testing MSE of KGHCG has the best convergence performance, followed by KGHCG-VP, and its stability gradually shows over time.
- (2) Table 5 summarizes the simulation results by different algorithms for the ENSO time series. Compared with the KGHCG algorithm, the testing time of KGHCG-VP significantly slumped by 66.3%. Although the prediction accuracy is slightly decreased, it is worth sacrificing tiny precision for a significant speedup in online tasks. In other words, KGHCG-VP has both low computational cost and high prediction accuracy. The prediction curves in Fig. 7 further confirm the high adaptability of the proposed algorithms on the ENSO dataset.

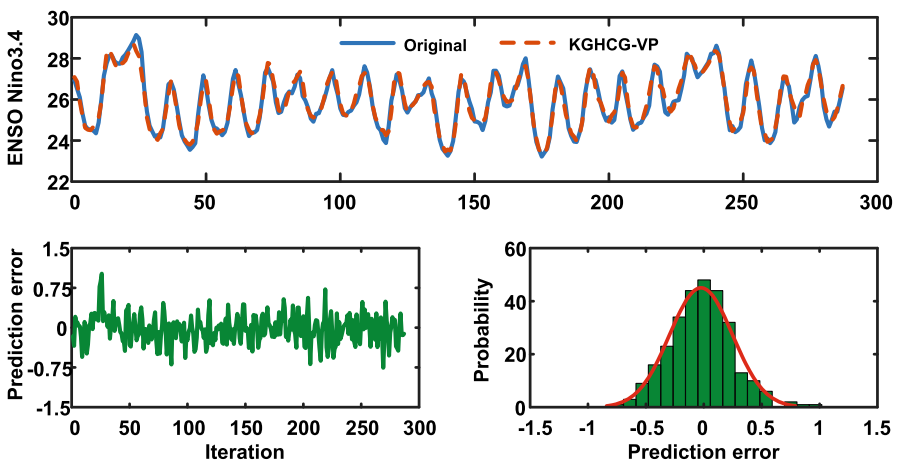
**Table 5** Simulation results by different methods for ENSO time series

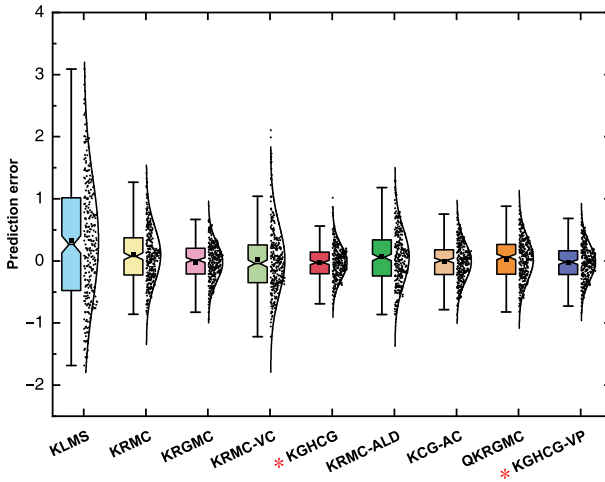
Methods		Training time	Testing time	RMSE	SMAPE	$R^2$	Dictionary
Baseline	KLMS [20]	1.6823	0.0129	1.0510	0.0324	0.3350	1150
	KRMC [35]	5.2324	0.0133	0.4536	0.0135	0.8761	1150
	KRGMC [39]	7.7886	0.0121	0.3019	0.0093	0.9451	1150
	KRMC-VC [22]	5.2016	0.0116	0.5576	0.0157	0.8128	1150
	KGHCG	5.3470	0.0089	0.2725	0.0082	0.9553	1150
Sparsity	KRMC-ALD	0.3770	0.0049	0.5018	0.0150	0.8484	180
	KCG-AC [37]	0.2971	0.0032	0.3044	0.0093	0.9442	146
	QKRGMC [30]	0.2295	0.0033	0.3421	0.0109	0.9295	146
	KGHCG-VP	0.2763	0.0030	0.2905	0.0087	0.9492	133

- (3) Error boxplots and scatter plots are drawn after each iteration, respectively, and the prediction accuracy is visually judged from Figs. 8 and 9. What is clearly presented in Fig. 8 is that the error range and the normal distribution curve of the proposed KGHCG and KGHCG-VP are small and concentrated. In addition, although the regression lines and baselines of KRGMC [39], KGHCG, KCG-AC [37], and KGHCG-VP have a small gap, KGHCG has the highest prediction accuracy, followed by KGHCG-VP when the  $R^2$  index and scatter distribution are considered at the same time.

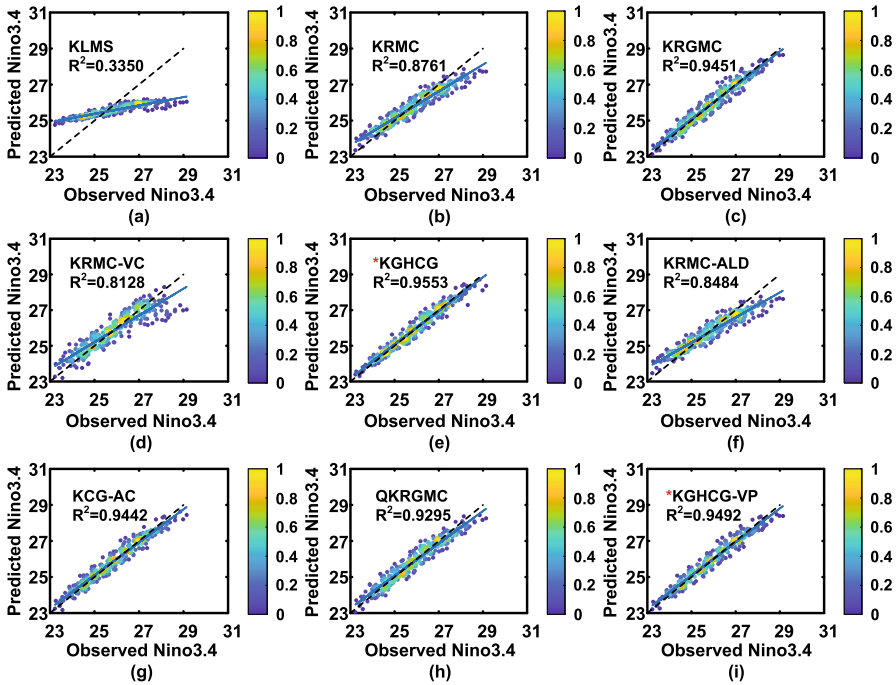
### 4.3 Beijing Air Quality Time Series

To further validate the efficacy of the proposed KGHCG-VP algorithm on large samples, we design multi-step prediction experiments using different sparse algorithms.

**Fig. 7** Prediction curves and error distribution by KGHCG-VP for ENSO time series



**Fig. 8** Boxplots and distributions of prediction errors by different methods for ENSO time series. The left side is the error boxplot, and the right side is the normal curve of error distributions. The black rectangle represents the median error. The shorter the height of the boxplot on the left, the closer the median is to zero, the smaller the vertical range of the error distribution curve on the right, and the fatter the horizontal range near the zero value, indicating the higher the prediction accuracy of the model



**Fig. 9** Scatter plots of predicted and observed Nino3.4 by different methods for ENSO time series. The solid blue line is the regression line, and the dashed black line is the 1:1 baseline. The smaller the angle between the regression line and the baseline, the higher the prediction accuracy of the model

In practical meteorology, the changes of atmospheric pollution concentrations are highly nonlinear and chaotic, and accurate prediction of them has an important effect on the improvement of human health and environmental quality. Therefore, we use the Beijing air quality time series to forecast PM2.5 concentration. Among them, the input variables are hourly PM2.5, PM10, SO2, NO2, O3, and CO in 2021. The output target is PM2.5 index and the horizons are one-step, five-step, and ten-step prediction. Through PSRT, the embedded dimension and delay time are calculated as [3,3,3,3,4,4] and [10,10,10,10,8,10], respectively. After that, we select 80% of the reconstructed 8000 datasets as the training set and the rest as the testing set.

For the Beijing air quality time series, the parameters setting of different sparse algorithms is shown in Table 6. Table 7 summarizes simulation results for multi-step-ahead PM2.5 prediction. Obviously, compared with the KGHCG algorithm that loads complete dictionaries into the model, KGHCG-VP effectively filters the input data according to the sparsification strategy. Finally, a compact dictionary of size 250 is generated, which greatly improves computational efficiency and facilitates fast real-time prediction. For comparison of prediction precision, RMSE and  $R^2$  of KGHCG-VP are consistently excellent. On the contrary, the SMAPE indicator performs poorly. Therefore, we further calculate the change rate between different horizons of SMAPE and obtain the conclusion that the change rate of KGHCG-VP from five step to ten step is the smallest (12.17%). Combining the testing time, RMSE, and  $R^2$ , KGHCG-VP can still maintain high prediction accuracy while shortening the operation time.

In addition, we plot the variation trend of  $R^2$  under different steps in Fig. 10. It is intuitively observed that KGHCG-VP has the highest prediction accuracy for one-step prediction, and the change between different steps is the smallest. By calculating the change rates of  $R^2$  regarding all sparse algorithms, the change rates of KGHCG-VP are the smallest, which are 23.84% and 55.82%, respectively. The numerical results are consistent with the observed results, which proves the superiority of the proposed algorithm.

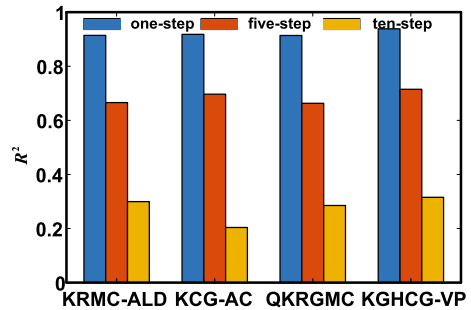
Finally, we also visualize the learning curves of different sparse algorithms for multi-step prediction of the PM2.5 indicator in Beijing, and we conclude from Fig. 11 that the prediction accuracy gradually decreases as the prediction range increases, but the fitted curve of KGHCG-VP is still superior to the competitors. Especially in the prediction of ten-step, KGHCG can effectively track the changes of time series, which further demonstrates the efficiency of the algorithm.

**Table 6** Parameters setting of different methods for Beijing air quality time series ( $\eta$  is the step parameter.  $\gamma$  is the regularization factor.  $\sigma'$  is the bandwidth of the correntropy.  $\alpha$  is the shape factor.  $\beta$  is the scale factor.  $c$  is the center location of the kernel.  $L$  is the number of sliding data.  $\nu$ ,  $\xi$  and  $\tau$  are the predefined threshold.  $\varepsilon$  is the quantization size)

Methods	$\eta$	$\gamma$	$\sigma'$	$\alpha$	$\beta$	$c$	$L$	$\nu$	$\xi$	$\varepsilon$	$\tau$	Kernel type	Kernel width
KRMC-ALD	–	$1e-5$	300	–	–	–	–	0.45	–	–	–	Gauss	135
KCG-AC [37]	–	–	–	–	–	–	–	–	0.82	–	–	Gauss	135
QKRGMC [30]	–	3.4	–	1.25	0.5	–	–	–	–	100	–	Gauss	135
KGHCG-VP	–	–	–	1.25	0.5	–	–	–	–	–	0.80	Gauss	135

**Table 7** Simulation results of different sparsity methods for Beijing air quality time series

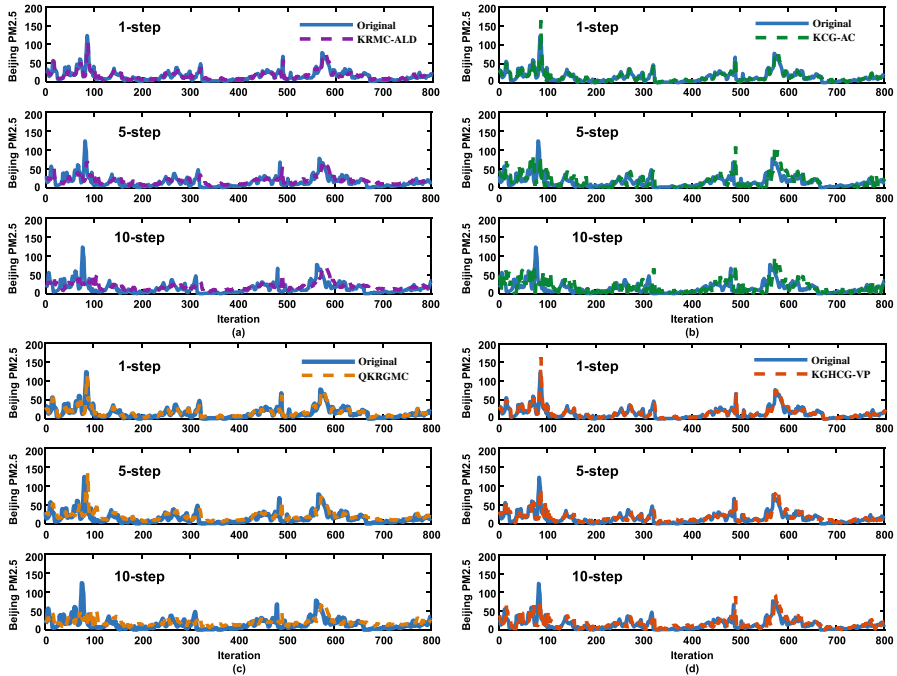
Step	Methods	Training time	Testing time	RMSE	SMAPE	$R^2$	Dictionary
One-step	KRMC-ALD	15.9589	0.0219	9.6948	0.4012	0.9142	212
	KCG-AC [37]	16.6106	0.0234	9.4599	0.5149	0.9183	271
	QKRGMC [30]	15.3835	0.0228	9.7071	0.3819	0.9140	207
	KGHCG-VP	14.7651	0.0202	8.2007	0.4641	0.9386	250
Five-step	KRMC-ALD	15.7458	0.0217	19.1356	0.5344	0.6657	212
	KCG-AC [37]	16.4261	0.0260	18.2173	0.7764	0.6970	271
	QKRGMC [30]	15.3570	0.0213	19.2038	0.5310	0.6633	207
	KGHCG-VP	15.0203	0.0200	17.6752	0.6942	0.7148	250
Ten-step	KRMC-ALD	15.6174	0.0220	27.6992	0.6411	0.2996	212
	KCG-AC [37]	16.3829	0.0250	29.5268	0.8852	0.2041	271
	QKRGMC [30]	15.3380	0.0220	27.9799	0.6531	0.2853	207
	KGHCG-VP	14.4896	0.0204	27.3769	0.7787	0.3158	250

**Fig. 10** The  $R^2$  index of different sparse methods for multi-step prediction of PM2.5 concentration in Beijing

## 5 Conclusion

This paper proposes a robust KGHCG algorithm. To be specific, our work is the first to develop a GHC criterion applied to KAF, which utilizes HQ optimization to cope with the nonconvex property of the GCC function. After that, the usage of the CG method has a positive effect on the convergence performance and prediction accuracy of KGHCG. In addition, the KGHCG-VP algorithm using the sparse strategy can resist the non-Gaussian noise while reducing the computational burden significantly. Experimental results show that KGHCG and its sparse variants have superior performance for online prediction tasks.

In our future work, we will consider the combination of KAF and evolving fuzzy systems to design an approach with strong structural adaptation and excellent adaptability in non-Gaussian noise environments. In addition, proper sparsification methods will also enhance the online prediction performance of the model, which is worthy of further study.



**Fig. 11** Prediction curves of different sparsity methods for multi-step prediction of PM<sub>2.5</sub> concentration in Beijing

**Acknowledgements** This work was supported by the National Natural Science Foundation of China (62173063).

**Funding** This work was supported by the National Natural Science Foundation of China (62173063).

**Data Availability** The datasets used to support the findings of this study are available from the corresponding author on reasonable request.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interests.

**Consent for publication** Not applicable.

**Ethics approval and consent to participate** Not applicable.

## References

1. F. Albu, K. Nishikawa, The kernel proportionate NLMS algorithm, in *21st European Signal Processing Conference (EUSIPCO 2013)* (IEEE, 2013), pp. 1–5
2. F. Albu, K. Nishikawa, A fixed budget implementation of a new variable step size kernel proportionate NLMS algorithm, in *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)* (IEEE, 2014), pp. 890–894

3. F. Albu, K. Nishikawa, New iterative kernel algorithms for nonlinear acoustic echo cancellation, in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)* (IEEE, 2015), pp. 734–739
4. F. Albu, K. Nishikawa, Low complexity kernel affine projection-type algorithms with a coherence criterion, in *2017 International Conference on Signals and Systems (ICSigSys)* (IEEE, 2017), pp. 87–91
5. P.S. Chang, A.N. Willson, Analysis of conjugate gradient algorithms for adaptive filtering. *IEEE Trans. Signal Process.* **48**(2), 409–418 (2000)
6. B. Chen, L. Xing, H. Zhao, N. Zheng, J.C. Pri et al., Generalized correntropy for robust adaptive filtering. *IEEE Trans. Signal Process.* **64**(13), 3376–3387 (2016)
7. I. Dassios, Analytic loss minimization: theoretical framework of a second order optimization method. *Symmetry* **11**(2), 136 (2019)
8. I. Dassios, D. Baleanu, Optimal solutions for singular linear systems of caputo fractional differential equations. *Math. Methods Appl. Sci.* **44**(10), 7884–7896 (2021)
9. I. Dassios, K. Fountoulakis, J. Gondzio, A preconditioner for a primal-dual newton conjugate gradient method for compressed sensing problems. *SIAM J. Sci. Comput.* **37**(6), A2783–A2812 (2015)
10. Y. Engel, S. Mannor, R. Meir, The kernel recursive least-squares algorithm. *IEEE Trans. Signal Process.* **52**(8), 2275–2285 (2004)
11. S. Garcia-Vega, X. Zeng, J. Keane, Stock returns prediction using kernel adaptive filtering within a stock market interdependence approach. *Expert Syst. Appl.* **160**, 113668 (2020)
12. Y. He, F. Wang, Y. Li, J. Qin, B. Chen, Robust matrix completion via maximum correntropy criterion and half-quadratic optimization. *IEEE Trans. Signal Process.* **68**, 181–195 (2020)
13. A.R. Heravi, G.A. Hodtani, A new information theoretic relation between minimum error entropy and maximum correntropy. *IEEE Signal Process. Lett.* **25**(7), 921–925 (2018)
14. F. Huang, J. Zhang, S. Zhang, Maximum versoria criterion-based robust adaptive filtering algorithm. *IEEE Trans. Circuits Syst. II Express Briefs* **64**(10), 1252–1256 (2017)
15. A. Khalili, A. Rastegarnia, M.K. Islam, T.Y. Rezaei, Steady-state tracking analysis of adaptive filter with maximum correntropy criterion. *Circuits Syst. Signal Process.* **36**(4), 1725–1734 (2017)
16. M.K. Khandani, W.B. Mikhael, Effect of sparse representation of time series data on learning rate of time-delay neural networks. *Circuits Syst. Signal Process.* **40**(4), 1–26 (2021)
17. D. Li, M. Han, J. Wang, Chaotic time series prediction based on a novel robust echo state network. *IEEE Trans. Neural Netw. Learn. Syst.* **23**(5), 787–799 (2012)
18. D. Liu, H. Zhao, X. He, L. Zhou, Polynomial constraint generalized maximum correntropy normalized subband adaptive filter algorithm. *Circuits Syst. Signal Process.* **41**, 1–18 (2021)
19. W. Liu, P.P. Pokharel, J.C. Principe, Correntropy: properties and applications in non-Gaussian signal processing. *IEEE Trans. Signal Process.* **55**(11), 5286–5298 (2007)
20. W. Liu, P.P. Pokharel, J.C. Principe, The kernel least-mean-square algorithm. *IEEE Trans. Signal Process.* **56**(2), 543–554 (2008)
21. W. Liu, J.C. Principe, S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction* (Wiley, New York, 2011)
22. X. Liu, C. Song, Z. Pang, Kernel recursive maximum correntropy with variable center. *Signal Process.* **191**, 108364 (2022)
23. V.J. Mathews, S.H. Cho, Improved convergence analysis of stochastic gradient adaptive filters using the sign algorithm. *IEEE Trans. Acoust. Speech Signal Process.* **35**(4), 450–454 (1987)
24. C.C. Paige, M.A. Saunders, LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw. (TOMS)* **8**(1), 43–71 (1982)
25. B. Ramadevi, K. Bingi, Chaotic time series forecasting approaches using machine learning techniques: a review. *Symmetry* **14**(5), 955 (2022)
26. C. Richard, J.C.M. Bermudez, P. Honeine, Online prediction of time series data with kernels. *IEEE Trans. Signal Process.* **57**(3), 1058–1067 (2008)
27. S. Ruder, An overview of gradient descent optimization algorithms. arXiv preprint [arXiv:1609.04747](https://arxiv.org/abs/1609.04747) (2016)
28. S. Sankar, A. Kar, S. Burra, M. Swamy, V. Mladenovic, Nonlinear acoustic echo cancellation with kernelized adaptive filters. *Appl. Acoust.* **166**, 107329 (2020)
29. T. Shao, Y.R. Zheng, J. Benesty, An affine projection sign algorithm robust against impulsive interferences. *IEEE Signal Process. Lett.* **17**(4), 327–330 (2010)



30. T. Shen, W. Ren, M. Han, Quantized generalized maximum correntropy criterion based kernel recursive least squares for online time series prediction. *Eng. Appl. Artif. Intell.* **95**, 103797 (2020)
31. F. Tan, X. Guan, Research progress on intelligent system 's learning, optimization, and control—part II: online sparse kernel adaptive algorithm. *IEEE Trans. Syst. Man Cybern. Syst.* **50**(12), 5369–5385 (2020)
32. G.K. Vallis, El niño: A chaotic dynamical system? *Science* **232**(4747), 243–245 (1986)
33. H. Wang, X. Li, D. Bi, X. Xie, Y. Xie, A robust student's t-based kernel adaptive filter. *IEEE Trans. Circuits Syst. II Express Briefs* **68**(10), 3371–3375 (2021)
34. W. Wang, H. Zhao, B. Chen, Robust adaptive volterra filter under maximum correntropy criteria in impulsive environments. *Circuits Syst. Signal Process.* **36**(10), 4097–4117 (2017)
35. Z. Wu, J. Shi, X. Zhang, W. Ma, B. Chen, I. Senior Member, Kernel recursive maximum correntropy. *Signal Process.* **117**, 11–16 (2015)
36. K. Xiong, H.H. Iu, S. Wang, Kernel correntropy conjugate gradient algorithms based on half-quadratic optimization. *IEEE Trans. Cybern.* **51**(11), 5497–5510 (2020)
37. M. Zhang, X. Wang, X. Chen, A. Zhang, The kernel conjugate gradient algorithms. *IEEE Trans. Signal Process.* **66**(16), 4377–4387 (2018)
38. C. Zhao, W. Ren, M. Han, Adaptive sparse quantization kernel least mean square algorithm for online prediction of chaotic time series. *Circuits Syst. Signal Process.* **40**(9), 4346–4369 (2021)
39. J. Zhao, H. Zhang, Kernel recursive generalized maximum correntropy. *IEEE Signal Process. Lett.* **24**(12), 1832–1836 (2017)
40. J. Zhao, H. Zhang, G. Wang, Projected kernel recursive maximum correntropy. *IEEE Trans. Circuits Syst. II Express Briefs* **65**(7), 963–967 (2018)
41. K. Zhong, J. Ma, M. Han, Online prediction of noisy time series: dynamic adaptive sparse kernel recursive least squares from sparse and adaptive tracking perspective. *Eng. Appl. Artif. Intell.* **91**, 103547 (2020)
42. G. Zoutendijk, Nonlinear programming, computational methods, in *Integer & Nonlinear Programming* (1970), pp. 37–86

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.