

Essay

Detecting Structural Change Point in ARMA Models via Neural Network Regression and LSCUSUM Methods

Xi-hame Ri¹, Zhanshou Chen^{1,2,*}  and Yan Liang^{3,4}

¹ School of Mathematics and Statistic, Qinghai Normal University, Xining 810008, China

² The State Key Laboratory of Tibetan Intelligent Information Processing and Application, Xining 810008, China

³ School of Computer Science and Technology, Qinghai Normal University, Xining 810008, China

⁴ School of Preparatory Education for Nationalities, Ningxia University, Yinchuan 750002, China

* Correspondence: chenzhanshou@126.com

Abstract: This study considers the change point testing problem in autoregressive moving average (ARMA) (p, q) models through the location and scale-based cumulative sum (LSCUSUM) method combined with neural network regression (NNR). We estimated the model parameters via the NNR method based on the training sample, where a long AR model was fitted to obtain the residuals. Then, we selected the optimal model orders p and q of the ARMA models using the Akaike information criterion based on a validation set. Finally, we used the forecasting errors obtained from the selected model to construct the LSCUSUM test. Extensive simulations and their application to three real datasets show that the proposed NNR-based LSCUSUM test performs well.

Keywords: ARMA models; change point; neural networks regression; LSCUSUM test



Citation: Ri, X.-h.; Chen, Z.; Liang, Y. Detecting Structural Change Point in ARMA Models via Neural Network Regression and LSCUSUM Methods. *Entropy* **2023**, *25*, 133. <https://doi.org/10.3390/e25010133>

Academic Editors: Augustine Wong, Xiaoping Shi and Christian H. Weiss

Received: 1 December 2022

Revised: 2 January 2023

Accepted: 6 January 2023

Published: 9 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The change point problem has been extensively studied in time series analysis because time series often suffer from structural changes due to issues such as changes of governmental policy, health care quality, and critical social events. This problem has also been crucial in engineering, medicine, and economics, and numerous related articles have been published. We refer the reader to Csörgö and Horváth [1], Hušková et al. [2], Franke et al. [3], Kang and Lee [4], Lee and Chen [5], and Huh et al. [6] for a general review.

The cumulative sum (CUSUM) test proposed by Page [7] is one of the most popular methods for change point tests due to its convenience of usage. Lee and Na [8] proposed the conventional estimate-based CUSUM test, which generally performs well, but suffers from severe size distortions and produces low powers on some occasions. Na et al. [9] and Lee [10] suggested the residual-based CUSUM test for time series models as a remedy. This test is more stable, but it suffers serious power loss when location parameters change because it can only detect the change in scale parameters. To improve upon this shortcoming, Oh [11,12] proposed a score vector-based CUSUM test and a modified residual-based CUSUM method to enhance the power performance. Lee [13] introduced a location- and scale-based CUSUM (LSCUSUM) test, and simulations confirmed its improved performance in terms of stability and power over any estimate-, residual- and score vector-based CUSUM tests developed thus far.

A key step in constructing the LSCUSUM test is to accurately estimate the residuals, i.e., fitting a correct model is important if we want to test change points in time series. The most popular time series fitting method is the classical ARMA method, which uses maximum likelihood estimation (QMLE) or least squares estimation (LSE) to fit an ARMA model. Classical linear ARMA models produce an accurate prediction when a time series truly follows them, but the prediction is not correct if the time series has obvious nonlinear characteristics. Because many time series carry nonlinear characteristics, new analytical tools such as neural network regression (NNR), support vector regression (SVR), and least

absolute shrinkage and selection operator (LASSO) have been applied to fit them, and have been demonstrated to outperform classical time series models, especially when a time series has certain nonlinear and nonstationary characteristics. For surveys, we refer the reader to Hwang and Ang [14], Hossain et al. [15], Zafar et al. [16], and Giacomazzo [17]. Harchaoui and Lévy-Leduc [18] considered the change point estimation problem as a variable selection problem and used the LASSO method to estimate multiple change points in one-dimensional piecewise constant signals. Chan et al. [19] applied a group LASSO method to estimate multiple change points in a time series. Jin et al. [20] proposed a fast algorithm to estimate multiple change points in nonstationary time series models. Qian and Su [21] studied a multiple change points estimation problem for panel data models via an adaptive group fused LASSO. For more studies about change point estimation via a penalized model selection method, please refer to Lee et al. [22], Jin et al. [23], and Ciuperca and Maciak [24], among many others.

Lee et al. [25,26] proposed an SVR with a hybridization of the CUSUM method to detect change points in ARMA and generalized autoregressive conditional heteroscedastic (GARCH) models. As is well known, the NNR can approximate the nonlinearity of a time series without knowing the underlying dynamic structure by minimizing the empirical risk. Motivated by this, we adopted NNR to obtain ARMA residuals, and then constructed the LSCUSUM statistic to test change points. Our simulations show that the proposed method outperforms methods based on SVR, the adaptive least absolute shrinkage and selection operator (ALASSO), and classical ARMA.

The rest of this paper is organized as follows. Section 2 presents the principle of the LSCUSUM test for ARMA models. Section 3 introduces the NNR method in a general framework. Section 4 proposes a forecasting method based on the NNR fitting ARMA model, herein the NNR-ARMA model, and then explains how to apply the NNR-ARMA model to the construction of the LSCUSUM test. Section 5 performs Monte Carlo simulations to evaluate the validity of the proposed method under various linear and nonlinear ARMA models. Section 6 applies the method to three real datasets for illustration. Finally, Section 7 provides concluding remarks.

2. LSCUSUM Test for ARMA Models

The model discussed in this paper is

$$y_t = f(\Theta; y_{t-1}, \dots, y_{t-p}, \varepsilon_{t-1}, \dots, \varepsilon_{t-q}) + \varepsilon_t, \tag{1}$$

where f is an unknown function to be estimated, $\Theta = (\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q, \sigma^2)$ is an unknown parameter vector, p and q are real numbers, and ε_i are IID random variables with mean zero, covariance $\sigma^2 > 0$, and $E|\varepsilon_t|^4 < \infty$. For given observations y_1, \dots, y_n , we are interested in the null and alternative hypotheses

$$H_0 : \Theta_t = \Theta, \quad t = 1, \dots, n,$$

$$H_1 : \Theta_t = \begin{cases} \Theta, & t = 1, \dots, k \\ \Theta + \Delta, & t = k + 1, \dots, n' \end{cases}$$

where Θ is the unknown parameter vector, the nonzero constant vector Δ represents the parameter skip, and $k = [n\tau]$ is the location of the change point, where $\tau \in (0, 1)$ and $[\cdot]$ denotes the integral function. Model (1) includes AR, ARMA, threshold ARMA (TARMA) and time-varying AR (TVAR) models as special cases.

To test the above null and alternative hypotheses, we applied the following two LSCUSUM tests proposed by Lee [13]:

$$\hat{T}_n^{LS} = \max_{1 \leq k \leq n} \left\{ \frac{1}{n\hat{\gamma}_{1,n}^2} \left| \sum_{t=1}^k (y_t - \hat{\varepsilon}_t)\hat{\varepsilon}_t - \frac{k}{n} \sum_{t=1}^n (y_t - \hat{\varepsilon}_t)\hat{\varepsilon}_t \right|^2 + \frac{1}{n\hat{\gamma}_{2,n}^2} \left| \sum_{t=1}^k \hat{\varepsilon}_t^2 - \frac{k}{n} \sum_{t=1}^n \hat{\varepsilon}_t^2 \right|^2 \right\},$$

$$\hat{T}_n^{\max} = \max_{1 \leq k \leq n} \max \left\{ \frac{1}{\sqrt{n} \hat{\gamma}_{1,n}} \left| \sum_{t=1}^k (y_t - \hat{\varepsilon}_t) \hat{\varepsilon}_t - \frac{k}{n} \sum_{t=1}^n (y_t - \hat{\varepsilon}_t) \hat{\varepsilon}_t \right|, \frac{1}{\sqrt{n} \hat{\gamma}_{2,n}} \left| \sum_{t=1}^k \hat{\varepsilon}_t^2 - \frac{k}{n} \sum_{t=1}^n \hat{\varepsilon}_t^2 \right| \right\},$$

where $\hat{\varepsilon}_t = y_t - f(\hat{\Theta}; y_{t-1}, \dots, y_{t-p}, \varepsilon_{t-1}, \dots, \varepsilon_{t-q})$ denotes the residuals with $\hat{\Theta}$ as the estimator of Θ , and

$$\hat{\gamma}_{1,n}^2 = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{\varepsilon}_t)^2 \hat{\varepsilon}_t^2 - \left(\frac{1}{n} \sum_{t=1}^n (y_t - \hat{\varepsilon}_t) \hat{\varepsilon}_t \right)^2,$$

$$\hat{\gamma}_{2,n}^2 = \frac{1}{n} \sum_{t=1}^n \hat{\varepsilon}_t^4 - \left(\frac{1}{n} \sum_{t=1}^n \hat{\varepsilon}_t^2 \right)^2.$$

We rejected H_0 if $\hat{T}_n^{LS} > 2.4503$ or $\hat{T}_n^{\max} > 1.4596$ at the nominal 0.05 level. The critical values of these two tests were obtained through Monte Carlo simulations using two-dimensional standard Brownian motion.

Next, we introduce the NNR method in a general framework, propose a fitting method based on NNR to the ARMA model, and explain how to apply the NNR-ARMA method when constructing the LSCUSUM test.

3. Neural Network Regression

A neural network (NN) is a simplified model obtained by abstracting and modeling the human brain, which is a parallel connection of a set of nodes called neurons. It can effectively solve complex regression and classification problems with a large number of correlated variables and make accurate predictions for time series.

In this study, we define an NN with an input layer, a hidden layer, and an output layer. We consider a feedforward net with n input nodes, one layer of H hidden nodes, and one output node. As shown in Figure 1, n nodes representing independent variables on the left form the input layer, H nodes in the middle form the hidden layer, and the rightmost node belongs to the output layer representing dependent variables. These nodes are connected according to the arrows, and the number next to an arrow marks its weight.

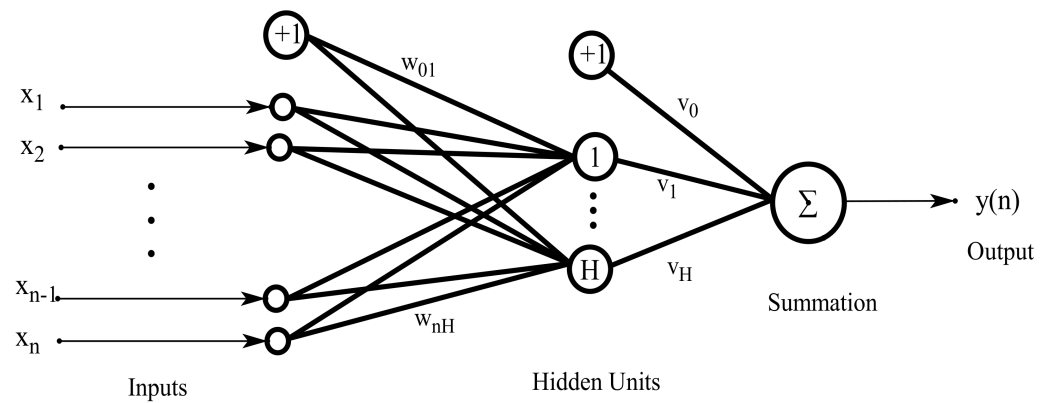


Figure 1. Proposed neural network topology.

Considering an input vector $\mathbf{x} = (x_1, \dots, x_n) \in \mathfrak{R}^n$, the output $z_h(\mathbf{x}; \theta)$ of the h -th hidden node is

$$z_h(\mathbf{x}; \theta) = \psi \left(\sum_{i=1}^n \omega_{ih} x_i + \omega_{0h} \right).$$

The output of the net is

$$F(\mathbf{x}; \boldsymbol{\theta}) = \psi^* \left(\sum_{h=1}^H v_h z_h(\mathbf{x}; \boldsymbol{\theta}) + v_0 \right) \\ = \psi^* \left\{ \sum_{h=1}^H v_h \left[\psi \left(\sum_{i=1}^n \omega_{ih} x_i + \omega_{0h} \right) \right] + v_0 \right\},$$

where ω_{ih} ($i = 0, \dots, n; h = 1, \dots, H$) is the weight connecting the i -th input layer node and h -th hidden layer node; v_h ($h = 0, \dots, H$) is the weight of the h -th hidden layer node to the j -th dependent variable; and $\boldsymbol{\theta}$ stands for all parameters v_0, \dots, v_H and ω_{ih} ($i = 0, \dots, n; h = 1, \dots, H$) of the network. We also write $\mathbf{v} = (v_0, \dots, v_H)^T$ and $\boldsymbol{\omega} = (\omega_{ih}, i = 0, \dots, n; h = 1, \dots, H)$. Here, $\psi(\cdot)$ and $\psi^*(\cdot)$ are activation functions of the NN, which are usually defined as S-shaped logistic functions,

$$\psi(x) = \frac{1}{1 + \exp(-ax)},$$

and

$$\lim \psi(x) = \begin{cases} 1, & x \rightarrow \infty \\ 0, & x \rightarrow -\infty \end{cases},$$

where a is a parameter that controls the inclination of the function. Different function types can be selected as required, such as the threshold, piecewise, Gaussian, and hyperbolic tangent functions.

The NN connection weights were adjusted through training, as follows:

1. Assume that the input vector and the associated output vector are $\mathbf{x} = \mathbf{x}_1, \dots, \mathbf{x}_m \in \mathfrak{R}^n$ and $\mathbf{y} = y_1, \dots, y_m \in \mathfrak{R}$, respectively.
2. Initialize the weights in the network:

$$\boldsymbol{\omega}^{(0)}, \mathbf{v}^{(0)}.$$

3. Use forward propagation to obtain the predicted value and calculate the loss function,

$$S(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) = \frac{1}{2} \sum_{j=1}^m (y_j - F(\mathbf{x}_j; \boldsymbol{\theta}))^2.$$

4. Update the connection weights:

$$\boldsymbol{\omega}^k = \boldsymbol{\omega}^{k-1} + \Delta \boldsymbol{\omega}; \\ \mathbf{v}^k = \mathbf{v}^{k-1} + \Delta \mathbf{v}$$

Taking individual weights, the k -th iteration weights are

$$\omega_{ih}^{(k)} = \omega_{ih}^{(k-1)} - \eta \left\{ \frac{\partial S(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}^{(k)})}{\partial \omega_{ih}} \right\}, \text{ for } i = 0, \dots, n, h = 1, \dots, H \text{ and } j = 1, \dots, m.$$

Similarly,

$$v_h^{(k)} = v_h^{(k-1)} - \eta \left\{ \frac{\partial S(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}^{(k)})}{\partial v_h} \right\}, \text{ for } h = 1, \dots, H \text{ and } j = 1, \dots, m.$$

where η ($0 < \eta < 1$) is the learning rate; k is the number of updates or iterations; and $\frac{\partial S(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}^{(k)})}{\partial \omega_{ih}}$ and $\frac{\partial S(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}^{(k)})}{\partial v_h}$ are respective error terms of the hidden unit and output unit,

i.e., the gradient values or partial derivatives of the loss function with respect to them. According to the chain rule, they are:

$$\frac{\partial S(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}^{(k)})}{\partial \omega_{hj}} = \frac{\partial S}{\partial \omega_{hj}} = \frac{\partial S}{\partial F} \frac{\partial F}{\partial z} \frac{\partial z}{\partial \omega_{hj}}$$

$$\frac{\partial S(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}^{(k)})}{\partial v_h} = \frac{\partial S}{\partial v_h} = \frac{\partial S}{\partial F} \frac{\partial F}{\partial z} \frac{\partial z}{\partial v_h}$$

- Repeat steps 3 and 4 until the loss function is less than the preset threshold or the number of iterations is exhausted and the output parameter is the best parameter at present.

An NN can have multiple hidden layers, but one is generally enough. The number of nodes in the hidden layer can be large or small. Too many nodes may lead to overfitting, and too few to poor fitting. The number of neurodes H in the hidden layer can be determined as in Looney [27] by rule of thumb as

$$H = \lceil 1.7 * \log_2(n) \rceil + 1.$$

Alternatively, one can use the Bayesian information criterion (BIC), as proposed by Swanson [28], to sequentially determine H . In this paper, the optimal number of nodes H is selected according to root mean square error (RMSE) recommendations through cross-validation.

4. Prediction Based on NNR-ARMA Model

Throughout this section, we assume that $\{y_t : t = 1, \dots, n, n + 1, \dots, n + l\}$ is generated according to the ARMA model (1). If the training sample is known to follow an NNR-ARMA (p, q) model with specific orders p and q , we can use this model. Otherwise, we determine the orders from the training sample as follows:

Step 1 Let $\mathbf{y} = (y_m, \dots, y_n)$, and design the matrix

$$\mathbf{X} = (X_m, X_{m+1}, \dots, X_n)^T$$

$$= \begin{pmatrix} y_{m-1} & \cdots & y_{m-p} & \varepsilon_{m-1} & \cdots & \varepsilon_{m-q} \\ y_m & \cdots & y_{m-p+1} & \varepsilon_m & \cdots & \varepsilon_{m-q+1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ y_{n-1} & \cdots & y_{n-p} & \varepsilon_{n-1} & \cdots & \varepsilon_{n-q} \end{pmatrix}$$

as the NN input. The design matrix \mathbf{X} involves the potential innovation terms $\varepsilon_t (t = m - q, \dots, n - 1)$. Thus, the long $AR(p')$ model is first fitted to y_1, \dots, y_n to obtain the residual $\hat{\varepsilon}_t$, and then the residual $\hat{\varepsilon}_t$ is used to approximate the unobserved ε_t . See Hannan and Rissanen [29] for more details. The design matrix with ε_t replaced is $\hat{\mathbf{X}}$, where $m = p' + \max(p, q) + 1$. When fitting the long $AR(p')$ model, the order p' can be determined using the Akaike information criterion (AIC), with the maximum order set to be $10 \log 10(n)$.

Step 2 Fix $p^* > 0, q^* > 0$, and the underlying true ARMA orders $p \leq p^*$ and $q \leq q^*$, where p^*, q^* are known upper bounds of the true orders. For each p, q , we assume that y_1, \dots, y_n approximately satisfy

$$y_t = f(y_{t-1}, \dots, y_{t-p}, \hat{\varepsilon}_{t-1}, \dots, \hat{\varepsilon}_{t-q}) + \hat{\varepsilon}_t, \tag{2}$$

and the NNR-ARMA (p, q) model is fitted with $\mathbf{y} = (y_m, \dots, y_n)$ and $\hat{\mathbf{X}}$, as in Section 3. Then, after applying the obtained NNR-ARMA (p, q) model to y_{n+1}, \dots, y_{n+l} , calculate the prediction errors $\hat{\varepsilon}_t (t = 1, \dots, l)$ and the corresponding AIC, namely, $AIC = 2(p + q) + l \ln(RSS/l)$,

where l denotes the number of effective observations and RSS denotes the sum of squared residuals. We then select the optimal order (p_0, q_0) by minimizing the AICs.

Step 3 Train the NNR-ARMA (p_0, q_0) model for the whole training sample y_1, \dots, y_{n+l} and predict the testing samples with the estimated NNR-ARM (p_0, q_0) model. The obtained prediction errors are then used in the construction of the NNR-based LSCUSUM test \hat{T}_n^{LS} and \hat{T}_n^{\max} .

5. Monte Carlo Simulation

We conducted Monte Carlo simulations to evaluate the finite sample performance of the NNR-based LSCUSUM test for linear and nonlinear ARMA models. For this, we used the AR, ARMA, TARMA, and TVAR models to generate data, and focused on the comparison of NNR-, ALASSO-, SVR-, and classical ARMA-based LSCUSUM tests. For each simulation, the empirical sizes and powers were calculated as the number of rejections of the null hypothesis of no changes out of 1000 repetitions at the 5% significance level. The sample sizes under consideration were $n = 300, 500,$ and 1000 . The change point was assumed to occur at $[n/2]$ under the alternatives. The simulations were conducted with R version 4.1.3, and we used the R packages nnet and caret for NNR, glmnet for ALASSO, and e1071 for SVR.

(1) Linear ARMA model

We generated time series from the ARMA (1,1) model:

$$y_t = \phi y_{t-1} + \theta \varepsilon_{t-1} + \varepsilon_t,$$

where $|\phi| < 1$ and ε_t are IID normal random variables with mean zero and variance σ^2 . We considered the following setups for the null hypothesis:

Model 1: AR(1) with $\phi = 0.3, \theta = 0,$ and $\sigma^2 = 1$.

Model 2: ARMA(1,1) with $\phi = 0.3, \theta = 0.3,$ and $\sigma^2 = 1$.

Under the alternative, we assumed a single parameter changed while the others remained constant. Tables 1 and 2 report the empirical sizes and powers for the AR(1) and ARMA(1,1) models, respectively, from which it can be seen that there was no size distortion, and reasonably good powers were produced using all four methods. In particular, the NNR-based LSCUSUM test outperformed the ALASSO-, SVR-, and classical ARMA-based LSCUSUM test, and for the LSCUSUM statistic, \hat{T}_n^{\max} was preferable to \hat{T}_n^{LS} in terms of stability and power. In addition, we conducted simulations with a sample size of 500 and different change point locations; see Table 3. It can be seen that the empirical power is greatly affected by the position of the change point, the NNR-based LSCUSUM test has the highest power when a change occurs in the middle, and the power decreases as the change point comes nearer to the beginning and ending points of time series, indicating that the change point is more easily detected when in the middle of the sequence.

Table 1. Empirical sizes and powers for AR(1) model with $\phi = 0.3, \sigma^2 = 1$ under the null and each alternative.

$\phi = 0.3, \sigma^2 = 1$		$n = 300$			$n = 500$			$n = 1000$					
		NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical
Size	\hat{T}_n^{LS}	0.056	0.044	0.038	0.043	0.046	0.045	0.047	0.046	0.061	0.044	0.044	0.045
	\hat{T}_n^{\max}	0.050	0.042	0.035	0.041	0.045	0.040	0.046	0.042	0.057	0.039	0.043	0.039
$\rightarrow \phi = 0.5$	\hat{T}_n^{LS}	0.283	0.267	0.226	0.268	0.458	0.457	0.419	0.449	0.814	0.809	0.788	0.810
	\hat{T}_n^{\max}	0.287	0.257	0.238	0.264	0.489	0.463	0.435	0.458	0.833	0.816	0.797	0.816
$\rightarrow \phi = 0.7$	\hat{T}_n^{LS}	0.929	0.876	0.818	0.872	0.999	0.992	0.983	0.992	1.000	1.000	1.000	1.000
	\hat{T}_n^{\max}	0.937	0.897	0.813	0.888	0.997	0.994	0.984	0.993	1.000	1.000	1.000	1.000
$\rightarrow \sigma^2 = 2$	\hat{T}_n^{LS}	0.908	0.903	0.885	0.916	1.000	0.999	0.997	0.999	1.000	1.000	1.000	1.000
	\hat{T}_n^{\max}	0.930	0.923	0.990	0.934	1.000	1.000	0.998	1.000	1.000	1.000	1.000	1.000

We also found that our method is obviously superior to the classical method when the change point occurs at the front of the sequence. Although not reported here, we carried

out simulations for various settings of model parameters and sample sizes, and a similar pattern was seen.

Table 2. Empirical sizes and powers for ARMA(1,1) model with $\phi = 0.3, \theta = 0.3, \sigma^2 = 1$ under the null and each alternative.

$\phi = 0.3, \theta = 0.3$ $\sigma^2 = 1$		$n = 300$			$n = 500$			$n = 1000$					
		NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical
Size	\hat{T}_n^{LS}	0.048	0.047	0.053	0.042	0.052	0.044	0.049	0.042	0.064	0.048	0.043	0.043
	\hat{T}_n^{\max}	0.045	0.046	0.047	0.038	0.041	0.048	0.046	0.042	0.063	0.046	0.043	0.042
→ $\phi = 0.5$	\hat{T}_n^{LS}	0.316	0.281	0.254	0.252	0.512	0.473	0.420	0.439	0.863	0.830	0.790	0.805
	\hat{T}_n^{\max}	0.316	0.279	0.251	0.242	0.522	0.474	0.408	0.440	0.870	0.850	0.794	0.824
→ $\phi = 0.7$	\hat{T}_n^{LS}	0.949	0.899	0.784	0.859	0.997	0.995	0.975	0.993	1.000	1.000	1.000	1.000
	\hat{T}_n^{\max}	0.954	0.917	0.802	0.869	0.998	0.998	0.981	0.996	1.000	1.000	1.000	1.000
→ $\theta = 0.7$	\hat{T}_n^{LS}	0.488	0.515	0.451	0.492	0.798	0.829	0.764	0.813	0.992	0.994	0.984	0.988
	\hat{T}_n^{\max}	0.519	0.527	0.451	0.512	0.810	0.837	0.774	0.821	0.993	0.993	0.984	0.992
→ $\sigma^2 = 2$	\hat{T}_n^{LS}	0.938	0.923	0.880	0.922	0.999	0.998	0.992	0.998	1.000	1.000	1.000	1.000
	\hat{T}_n^{\max}	0.931	0.925	0.884	0.927	0.997	0.997	0.996	0.997	1.000	1.000	1.000	1.000

Table 3. Empirical powers when change point occurs at $[n\tau]$ with $n = 500$ for ARMA(1,1) model.

$\phi = 0.3, \theta = 0.3$ $\sigma^2 = 1$		$\tau = 0.25$			$\tau = 0.50$			$\tau = 0.75$					
		NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical
Size	\hat{T}_n^{LS}	0.278	0.228	0.212	0.107	0.512	0.473	0.420	0.439	0.368	0.281	0.257	0.248
	\hat{T}_n^{\max}	0.287	0.226	0.204	0.100	0.522	0.474	0.408	0.440	0.359	0.298	0.260	0.263
→ $\phi = 0.5$	\hat{T}_n^{LS}	0.899	0.827	0.657	0.389	0.997	0.995	0.975	0.993	0.947	0.945	0.910	0.935
	\hat{T}_n^{\max}	0.914	0.857	0.691	0.377	0.998	0.998	0.981	0.996	0.952	0.950	0.931	0.952
→ $\phi = 0.7$	\hat{T}_n^{LS}	0.600	0.676	0.509	0.375	0.798	0.829	0.764	0.813	0.551	0.494	0.451	0.458
	\hat{T}_n^{\max}	0.628	0.684	0.527	0.389	0.810	0.837	0.774	0.821	0.541	0.497	0.450	0.452
→ $\theta = 0.7$	\hat{T}_n^{LS}	0.834	0.809	0.757	0.439	0.999	0.998	0.992	0.998	0.964	0.961	0.940	0.962
	\hat{T}_n^{\max}	0.861	0.871	0.796	0.510	0.997	0.997	0.996	0.997	0.969	0.969	0.953	0.969

(2) Threshold ARMA model

We generated time series from the TARMA(1,1) model:

$$y_t = \begin{cases} \phi_1 y_{t-1} + \theta_1 \varepsilon_{t-1} + \varepsilon_t, & y_{t-1} \leq 1 \\ \phi_2 y_{t-1} + \theta_2 \varepsilon_{t-1} + \varepsilon_t, & y_{t-1} > 1 \end{cases}$$

where ε_t are IID normal random variables with mean zero and variance σ^2 . We considered the following setup for the null hypothesis:

Model 3: TARMA(1,1) with $\phi_1 = 0.1, \phi_2 = -0.5, \theta_1 = \theta_2 = 0.5$, and $\sigma^2 = 1$.

Model 4: TARMA(1,1) with $\phi_1 = 0.3, \phi_2 = -0.5, \theta_1 = \theta_2 = 0.5$, and $\sigma^2 = 1$.

Under the alternative, we assumed a single parameter changed while the others remained constant. Tables 4 and 5 list some empirical sizes and powers for models 3 and 4, respectively. The results show that the classical ARMA- and ALASSO-based LSCUSUM tests had severe size distortion, which became more severe as the sample size increased. The NNR- and SVR-based LSCUSUM did not have such a problem, and the NNR-based LSCUSUM test produced better powers than the SVR-based LSCUSUM test. As anticipated, the power increased as the sample size increased or the deviance between the two parameter sets before and after the change point became larger. The simulation results considering different change point positions were the same as before, i.e., the LSCUSUM test had the highest power when a change point occurred in the middle of the time series (Table 6).

(3) Time-varying AR model

We generated a time series from the period AR(1) (PAR) model:

$$y_t = \phi_t y_{t-1} + \varepsilon_t$$

where ϕ_t is periodic with period $t = 1, 2$ and $\varepsilon_t \sim N(0, \sigma^2)$. We considered the following setup for the null hypothesis:

Model 5: PAR (1) with $\phi_1 = -0.3, \phi_2 = -0.5$, and $\sigma^2 = 1$.
 Model 6: PAR (1) with $\phi_1 = 0.3, \phi_2 = 0.5$, and $\sigma^2 = 1$.

Table 4. Empirical sizes and powers for threshold ARMA(1,1) model with $\phi_1 = 0.1, \phi_2 = -0.5, \theta = \theta_1 = \theta_2 = 0.5, \sigma^2 = 1$ under the null and each alternative.

$\phi_1 = 0.1, \phi_2 = -0.5$ $\theta = 0.5, \sigma^2 = 1$		$n = 300$				$n = 500$				$n = 1000$			
		NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical
Size	\hat{T}_n^{LS}	0.057	0.070	0.039	0.073	0.040	0.086	0.038	0.096	0.041	0.083	0.037	0.097
	\hat{T}_n^{\max}	0.045	0.066	0.040	0.068	0.040	0.092	0.041	0.095	0.041	0.081	0.043	0.095
$\rightarrow \phi_1 = 0.5$	\hat{T}_n^{LS}	0.464	0.310	0.213	0.445	0.888	0.817	0.820	0.825	1.000	1.000	0.994	1.000
	\hat{T}_n^{\max}	0.510	0.327	0.321	0.475	0.909	0.769	0.748	0.756	1.000	1.000	0.997	1.000
$\rightarrow \phi_1 = 0.7$	\hat{T}_n^{LS}	0.998	0.975	0.684	0.977	1.000	1.000	0.925	1.000	1.000	1.000	1.000	1.000
	\hat{T}_n^{\max}	0.998	0.951	0.748	0.959	1.000	1.000	0.951	1.000	1.000	1.000	1.000	1.000
$\rightarrow \theta = -0.5$	\hat{T}_n^{LS}	0.524	0.361	0.323	0.712	0.747	0.391	0.481	0.810	0.888	0.630	0.753	0.946
	\hat{T}_n^{\max}	0.531	0.390	0.352	0.737	0.756	0.398	0.513	0.818	0.899	0.635	0.776	0.946
$\rightarrow \sigma^2 = 2$	\hat{T}_n^{LS}	0.904	0.913	0.868	0.913	0.995	0.989	0.997	0.991	1.000	1.000	1.000	1.000
	\hat{T}_n^{\max}	0.909	0.920	0.880	0.923	0.994	0.993	0.991	0.991	1.000	1.000	1.000	1.000

Table 5. Empirical sizes and powers for threshold ARMA(1,1) model with $\phi_1 = 0.3, \phi_2 = -0.5, \theta = \theta_1 = \theta_2 = 0.5, \sigma^2 = 1$ under the null and each alternative.

$\phi_1 = 0.3, \phi_2 = -0.5$ $\theta = 0.5, \sigma^2 = 1$		$n = 300$				$n = 500$				$n = 1000$			
		NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical
Size	\hat{T}_n^{LS}	0.065	0.094	0.034	0.100	0.069	0.137	0.050	0.137	0.045	0.134	0.044	0.134
	\hat{T}_n^{\max}	0.067	0.094	0.035	0.096	0.068	0.140	0.039	0.134	0.048	0.136	0.042	0.135
$\rightarrow \phi_1 = 0.7$	\hat{T}_n^{LS}	0.880	0.799	0.433	0.836	0.994	0.999	0.795	1.000	1.000	1.000	0.996	1.000
	\hat{T}_n^{\max}	0.876	0.781	0.528	0.841	0.993	0.995	0.857	0.995	1.000	1.000	0.998	1.000
$\rightarrow \phi_1 = 0.9$	\hat{T}_n^{LS}	1.000	1.000	0.488	0.996	1.000	1.000	0.701	1.000	1.000	1.000	1.000	1.000
	\hat{T}_n^{\max}	1.000	1.000	0.543	0.997	1.000	1.000	0.769	1.000	1.000	1.000	1.000	1.000
$\rightarrow \theta = -0.5$	\hat{T}_n^{LS}	0.225	0.113	0.090	0.430	0.264	0.140	0.118	0.549	0.284	0.218	0.160	0.678
	\hat{T}_n^{\max}	0.259	0.110	0.095	0.472	0.278	0.135	0.126	0.585	0.295	0.212	0.183	0.701
$\rightarrow \sigma^2 = 2$	\hat{T}_n^{LS}	0.904	0.896	0.852	0.903	0.993	0.898	0.987	0.991	1.000	1.000	1.000	1.000
	\hat{T}_n^{\max}	0.919	0.910	0.865	0.911	0.996	0.988	0.988	0.988	1.000	1.000	1.000	1.000

Table 6. Empirical power of the change point occurs at $[n\tau]$ and $n = 500$ for threshold ARMA(1,1).

$\phi_1 = 0.1, \phi_2 = -0.5$ $\theta = 0.5, \sigma^2 = 1$		$\tau = 0.25$				$\tau = 0.50$				$\tau = 0.75$			
		NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical
$\rightarrow \phi_1 = 0.5$	\hat{T}_n^{LS}	0.365	0.378	0.298	0.479	0.888	0.817	0.478	0.825	0.629	0.576	0.510	0.626
	\hat{T}_n^{\max}	0.358	0.357	0.302	0.496	0.909	0.769	0.489	0.756	0.635	0.559	0.516	0.616
$\rightarrow \phi_1 = 0.7$	\hat{T}_n^{LS}	0.700	0.769	0.575	0.826	1.000	1.000	0.925	1.000	0.989	0.952	0.922	0.960
	\hat{T}_n^{\max}	0.726	0.756	0.608	0.836	1.000	1.000	0.951	1.000	0.993	0.940	0.932	0.951
$\rightarrow \theta = -0.5$	\hat{T}_n^{LS}	0.986	0.975	0.298	1.000	0.747	0.391	0.481	0.810	0.756	0.354	0.544	0.891
	\hat{T}_n^{\max}	0.988	0.975	0.271	0.999	0.756	0.398	0.513	0.818	0.720	0.351	0.489	0.868
$\rightarrow \sigma^2 = 2$	\hat{T}_n^{LS}	0.810	0.826	0.778	0.836	0.995	0.989	0.997	0.991	0.962	0.949	0.938	0.949
	\hat{T}_n^{\max}	0.814	0.864	0.787	0.962	0.994	0.993	0.991	0.991	0.961	0.954	0.943	0.955

Under the alternative, we assumed a single parameter changed while the others remained constant. Tables 7 and 8 summarize the empirical sizes and powers, respectively, for the PAR model. The results show that no tests had size distortions, and the NNR-based LSCUSUM test outperformed the ALASSO-, SVR- and classical ARMA-based LSCUSUM tests of stability and power. As in cases 1 and 2, NNR-based \hat{T}_n^{\max} was superior to the other tests. Table 9 shows a similar pattern to cases 1 and 2.

Table 7. Empirical sizes and powers for PAR(1) model with $\phi_1 = -0.3, \phi_2 = -0.5, T = 2$ under the null and each alternative.

$\phi_1 = -0.3, \phi_2 = -0.5$ $T = 2$		$n = 300$				$n = 500$				$n = 1000$			
		NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical
Size	\hat{T}_n^{LS}	0.053	0.049	0.056	0.039	0.055	0.063	0.058	0.051	0.057	0.049	0.048	0.047
	\hat{T}_n^{\max}	0.052	0.047	0.046	0.038	0.060	0.053	0.055	0.049	0.052	0.050	0.046	0.056

Table 7. Cont.

$\phi_1 = -0.3, \phi_2 = -0.5$ $T = 2$		$n = 300$				$n = 500$				$n = 1000$			
		NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical
$\rightarrow \phi_1 = -0.5$	\hat{T}_n^{LS}	0.140	0.113	0.112	0.095	0.185	0.163	0.155	0.136	0.337	0.333	0.281	0.290
	\hat{T}_n^{\max}	0.134	0.113	0.104	0.097	0.175	0.162	0.153	0.134	0.343	0.340	0.305	0.299
$\rightarrow \phi_1 = -0.7$	\hat{T}_n^{LS}	0.347	0.291	0.291	0.247	0.567	0.502	0.468	0.445	0.875	0.870	0.831	0.834
	\hat{T}_n^{\max}	0.344	0.289	0.293	0.240	0.570	0.506	0.482	0.452	0.888	0.869	0.847	0.848
$\rightarrow \phi_2 = -0.7$	\hat{T}_n^{LS}	0.094	0.079	0.086	0.069	0.137	0.091	0.117	0.082	0.202	0.199	0.212	0.170
	\hat{T}_n^{\max}	0.086	0.074	0.079	0.062	0.116	0.092	0.111	0.075	0.186	0.186	0.199	0.162
$\rightarrow \phi_2 = 0.3$	\hat{T}_n^{LS}	0.704	0.576	0.430	0.712	0.952	0.898	0.775	0.950	1.000	1.000	0.995	1.000
	\hat{T}_n^{\max}	0.693	0.580	0.420	0.714	0.954	0.899	0.747	0.950	1.000	1.000	0.996	1.000

Table 8. Empirical sizes and powers for threshold PAR(1) model with $\phi_1 = 0.3, \phi_2 = 0.5, T = 2$ under the null and each alternative.

$\phi_1 = 0.3, \phi_2 = 0.5$ $T = 2$		$n = 300$				$n = 500$				$n = 1000$			
		NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical
Size	\hat{T}_n^{LS}	0.054	0.048	0.039	0.044	0.047	0.050	0.048	0.039	0.059	0.041	0.032	0.031
	\hat{T}_n^{\max}	0.056	0.039	0.036	0.039	0.046	0.053	0.044	0.046	0.059	0.039	0.041	0.034
$\rightarrow \phi_1 = 0.5$	\hat{T}_n^{LS}	0.125	0.107	0.095	0.090	0.190	0.154	0.136	0.138	0.329	0.327	0.276	0.289
	\hat{T}_n^{\max}	0.119	0.106	0.097	0.094	0.176	0.153	0.137	0.139	0.327	0.317	0.278	0.287
$\rightarrow \phi_1 = 0.7$	\hat{T}_n^{LS}	0.346	0.315	0.269	0.262	0.564	0.506	0.475	0.443	0.862	0.859	0.830	0.831
	\hat{T}_n^{\max}	0.333	0.299	0.268	0.244	0.583	0.511	0.476	0.457	0.878	0.872	0.841	0.845
$\rightarrow \phi_2 = 0.7$	\hat{T}_n^{LS}	0.111	0.081	0.087	0.065	0.140	0.086	0.100	0.083	0.237	0.178	0.196	0.140
	\hat{T}_n^{\max}	0.098	0.074	0.075	0.062	0.125	0.086	0.097	0.070	0.230	0.160	0.182	0.128
$\rightarrow \phi_2 = -0.3$	\hat{T}_n^{LS}	0.717	0.589	0.417	0.754	0.953	0.921	0.777	0.961	1.000	1.000	0.987	1.000
	\hat{T}_n^{\max}	0.722	0.580	0.413	0.738	0.950	0.924	0.767	0.964	1.000	1.000	0.987	1.000

Table 9. Empirical power of the change point occurs at $[n\tau]$ for PAR(1).

$\phi_1 = -0.3, \phi_2 = -0.5$ $T = 2$		$\tau = 0.25$				$\tau = 0.50$				$\tau = 0.75$			
		NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical	NNR	ALASSO	SVR	Classical
$\rightarrow \phi_1 = -0.5$	\hat{T}_n^{LS}	0.123	0.095	0.096	0.088	0.185	0.163	0.155	0.136	0.128	0.106	0.112	0.096
	\hat{T}_n^{\max}	0.118	0.095	0.093	0.083	0.175	0.162	0.153	0.134	0.125	0.121	0.119	0.105
$\rightarrow \phi_1 = -0.7$	\hat{T}_n^{LS}	0.301	0.214	0.218	0.181	0.567	0.502	0.468	0.445	0.392	0.346	0.318	0.308
	\hat{T}_n^{\max}	0.288	0.225	0.219	0.170	0.570	0.506	0.482	0.452	0.397	0.344	0.332	0.307
$\rightarrow \phi_2 = -0.7$	\hat{T}_n^{LS}	0.084	0.064	0.077	0.054	0.137	0.091	0.117	0.082	0.104	0.083	0.105	0.071
	\hat{T}_n^{\max}	0.082	0.062	0.084	0.053	0.116	0.092	0.111	0.075	0.091	0.086	0.084	0.074
$\rightarrow \phi_2 = 0.3$	\hat{T}_n^{LS}	0.496	0.318	0.270	0.608	0.952	0.898	0.775	0.950	0.792	0.751	0.613	0.746
	\hat{T}_n^{\max}	0.512	0.315	0.270	0.621	0.954	0.899	0.747	0.950	0.798	0.763	0.600	0.744

6. Empirical Applications

We illustrate our proposed NNR-based LSCUSUM test scheme by analyzing three sets of real data: the annual volume of discharge from the Nile River at Aswan in $10^8 m^3$ from 1871 to 1970, the weekly average oil prices of the USA from 1 September 2014 to 27 August 2018, and the weekly Hang Seng Index (HSI) of Hong Kong from 4 September 2011 to 26 Jun 2022, whose respective time series were obtained from the R package TSA and the websites www.macrotrends.net (accessed on 27 June 2022) and investing.com. We used 70% samples of each dataset for training and the remaining 30% for testing. Prior to fitting the NNR-ARMA model, we inspected the autocorrelation function (ACF) and partial autocorrelation function (PACF) of each time series, looking for irregular patterns of autocorrelations, as seen in Figure 2, where plot (a) shows that the ACFs and PACFs for the Nile River dataset support stationarity to a great extent, indicating that estimation via the NNR-ARMA model with this time series would not undermine the outcomes. In contrast, plots (b) and (c) show that oil prices and HSI are nonstationary time series, and we use the log-returns of the oil prices and HIS.

We applied the NNR-based LSCUSUM test to the Nile River data and obtained $\hat{T}_n^{LS} = 9.589$ and $\hat{T}_n^{\max} = 2.631$, which are larger than the respective critical values 2.4503 and 1.4596 at the 0.05 level. Thus, we rejected the null hypothesis of no changes. We calculated that both statistics detected the change point at 1899, which is the conclusion of Cobb [30], MacNeill et al. [31], and Wu and Zhao [32]. They believed that there is a change (decrease)

in year 1899, which may be due to the construction of a new dam at Aswan. See Figure 3, where the red vertical line indicates a change point. For comparison, we also applied the classical ARMA-based LSCUSUM test to the Nile River data. Here, we expect this method to misbehave due to the nonlinearity of the datasets (Keenan’s linearity test [33] is quite against the linearity assumption). We obtained $\hat{T}^{LS} = 2.271 < 2.4503$, which does not reject the null hypothesis, whereas $\hat{T}^{\max} = 1.468 > 1.4596$ rejects it, indicating that the change point occurs in 1917. The blue vertical line in Figure 3 shows that the change point for the dataset is quite away from the previously obtained one. This result demonstrates that the classical ARMA-based LSCUSUM can be severely damaged when a time series sample has a strong nonlinear feature.

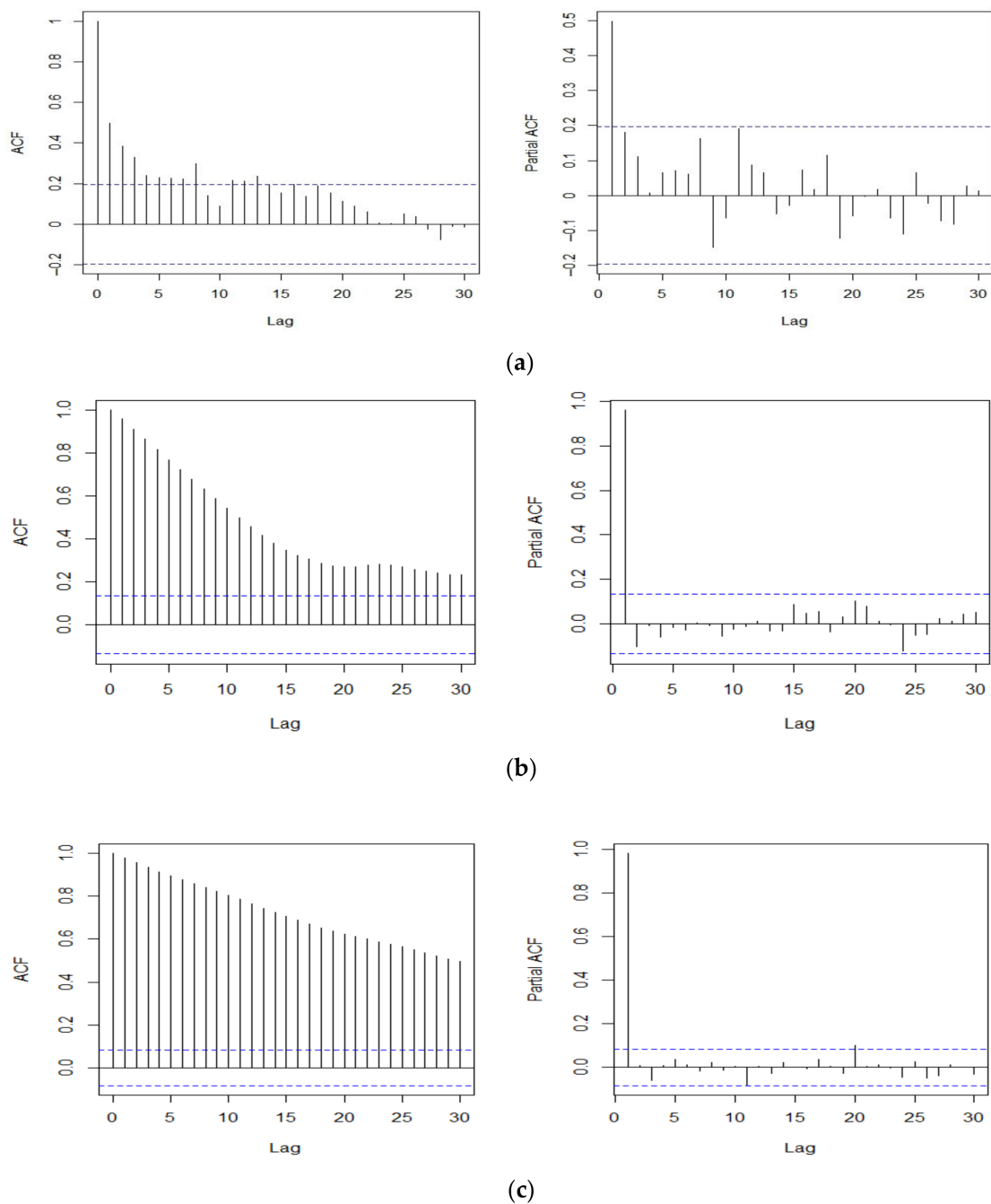


Figure 2. Plot of ACF and partial autocorrelation function (PACF) up to lag 30 of: (a) Nile River data; (b) oil prices; and (c) HSI.

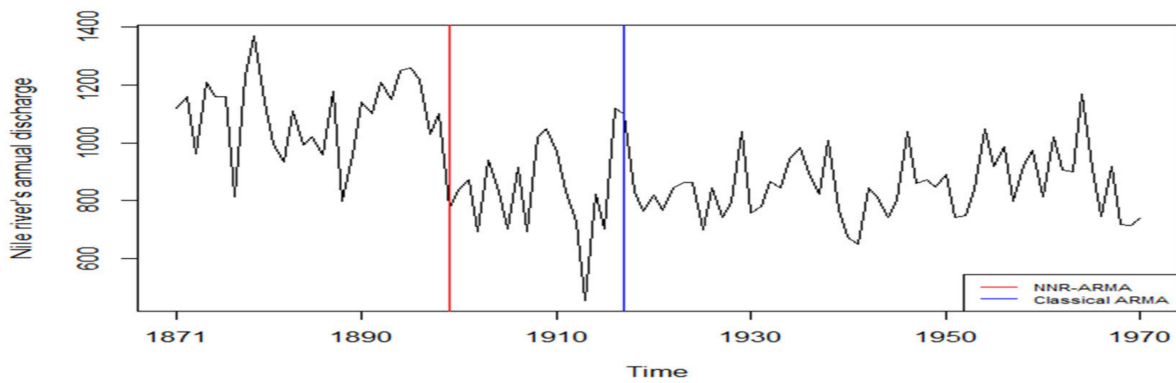
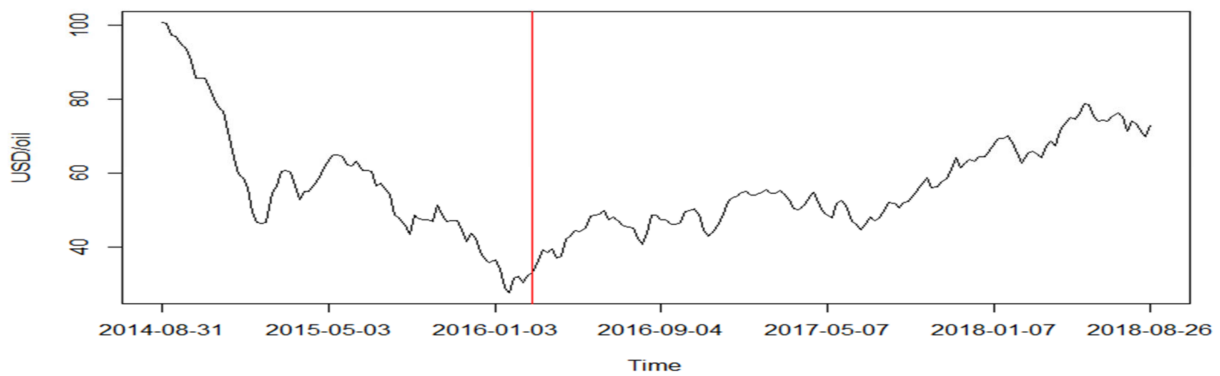
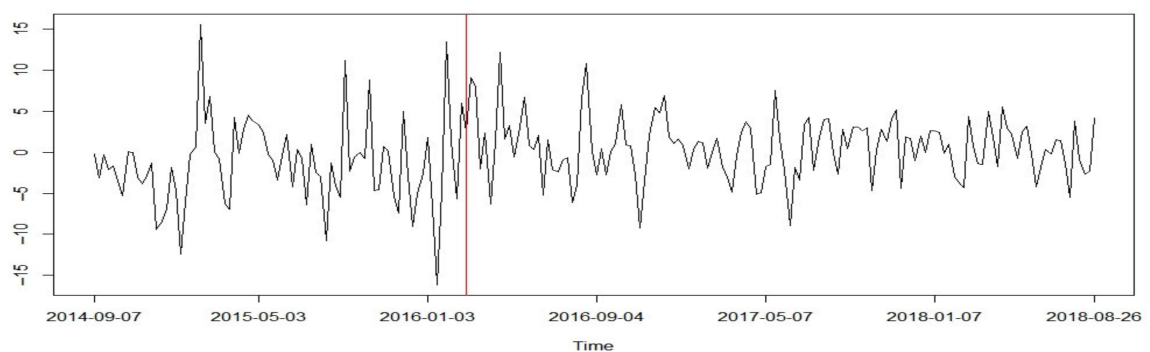


Figure 3. Annual volume of discharge from Nile River at Aswan, 1871–1970.

Furthermore, we obtained the NNR-based $\hat{T}^{LS} = 4.665$ and $\hat{T}^{\max} = 2.009$ for log-returns of oil prices, which indicates the detection of change in both cases at the 0.05 level. Both statistics identified the location of a change point at 14 February 2016, whereas we found that $\hat{T}^{LS} = 7.004$ and $\hat{T}^{\max} = 2.153$ for log-returns of HSI, and both statistics indicate that the change point occurs on 10 December 2017. Similar to the case of oil prices, a significant change in trend was observed in the original datasets, as shown in Figures 4 and 5.



(a)



(b)

Figure 4. (a) Weekly averages of oil prices; (b) log-returns with detected change point.

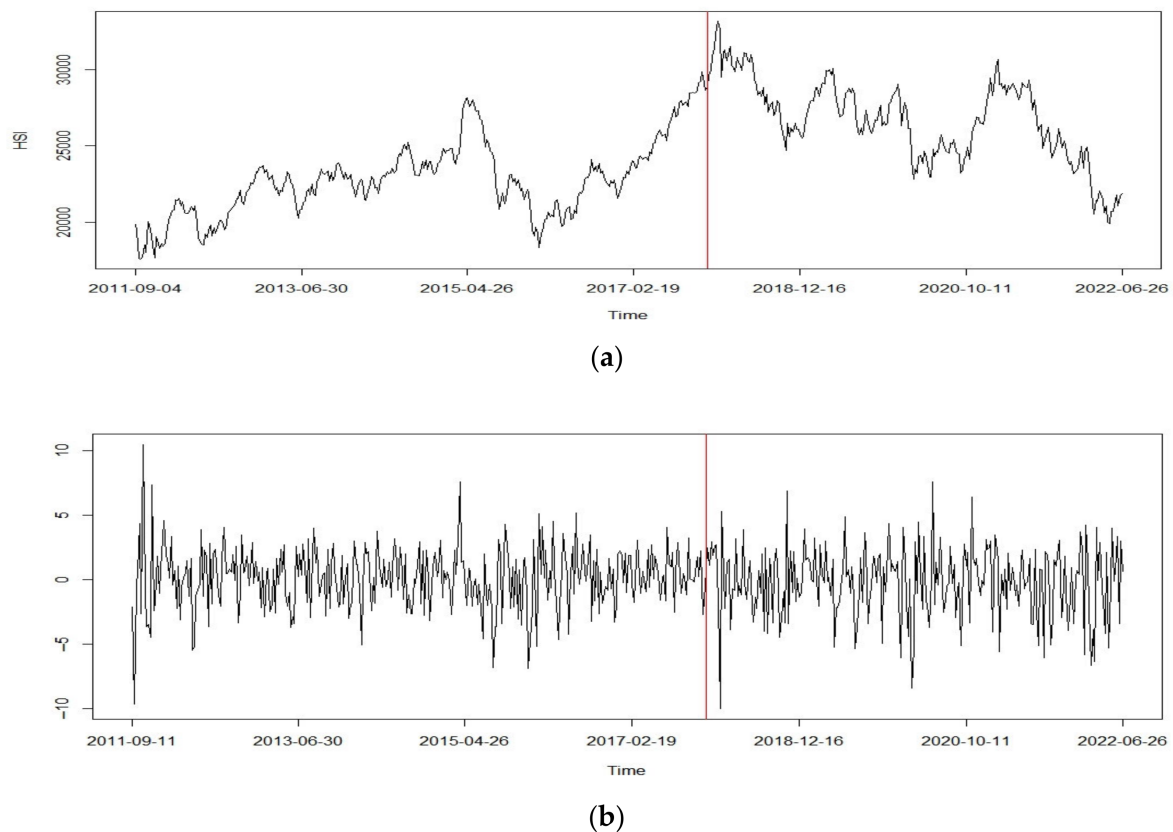


Figure 5. (a) Weekly price of HIS; (b) log-returns with detected change point.

7. Concluding Remarks

We proposed a prediction method based on the NNR-ARMA model and described how to determine an optimal model. This model was used to obtain residuals for a training time series sample, which was divided into two subseries. A long AR model was fitted to the first subseries to obtain the residuals, which were used as the error terms in the NNR-ARMA (p, q) model, and for each estimated NNR-ARMA (p, q) model, we calculated AICs based on the second subseries and selected an optimal ARMA order with the smallest AIC. Prediction errors or residuals were obtained using the determined NNR-ARMA (p, q) model and used to construct the LSCUSUM test. Monte Carlo simulations were conducted using the linear and nonlinear ARMA models with various parameter settings, including ARMA, TARMA, and TVAR. The results show that the NNR-based LSCUSUM test outperformed the classical ARMA- and ALASSO-based LSCUSUM tests, especially when the underlying model was nonlinear. Finally, our method was applied to the analysis of real datasets, namely of Nile River data, oil prices, and HSI prices, and detected one change point in all cases. Our findings support the validity of our method and its practicality in various real-world circumstances.

Author Contributions: Conceptualization, Z.C. and X.-h.R.; methodology, X.-h.R.; software, X.-h.R.; validation, X.-h.R. and Y.L.; formal analysis, Z.C.; writing—original draft preparation, X.-h.R.; writing—review and editing, Z.C. and Y.L.; visualization, X.-h.R. and Y.L.; supervision, Z.C.; project administration, Z.C.; funding acquisition, Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No.12161072), Natural Science Foundation of Qinghai Province (No.2019-ZJ-920).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Acknowledgments: The authors would like to thank the editor and the referees whose comments have improved the quality of this paper.

Conflicts of Interest: The authors declare no conflict interest.

References

1. Csörgö, M.; Horváth, L. *Limit Theorems in Change-Point Analysis*; John Wiley & Sons, Inc.: New York, NY, USA, 1997.
2. Hušková, M.; Prásková, Z.; Steinebach, J. On the detection of changes in autoregressive time series Asymptotic. *J. Stat. Plan. Inference* **2007**, *137*, 1243–1259. [[CrossRef](#)]
3. Franke, J.; Kirch, C.; Kamgaing, J.T. Change-points in times series of counts. *J. Time Ser. Anal.* **2012**, *33*, 757–770. [[CrossRef](#)]
4. Kang, J.; Lee, S. Parameter change test for Poisson autoregressive models. *Scand. J. Stat.* **2014**, *41*, 1136–1152. [[CrossRef](#)]
5. Lee, S.; Lee, Y.; Chen, C.W.S. Parameter change test for zero-inflated generalized Poisson autoregressive models. *Statistics* **2016**, *50*, 540–557. [[CrossRef](#)]
6. Huh, J.; Kim, H.; Lee, S. Monitoring parameter shift with Poisson integer-valued GARCH models. *J. Stat. Comput. Simul.* **2017**, *87*, 1754–1766. [[CrossRef](#)]
7. Page, E.S. A test for a change in a parameter occurring at an unknown point. *Biometrika* **1955**, *42*, 523–527. [[CrossRef](#)]
8. Lee, S.; Na, O.; Na, S. On the cusum of squares test for variance change in nonstationary and nonparametric time series models. *Ann. Inst. Stat. Math.* **2003**, *55*, 467–485. [[CrossRef](#)]
9. Na, O.; Lee, J.; Lee, S. Change point detection in SCOMDY models. *AStA Adv. Stat. Anal.* **2013**, *97*, 215–238. [[CrossRef](#)]
10. Lee, J.; Lee, S. Parameter change test for nonlinear time series models with GARCH type errors. *J. Korean Math. Soc.* **2015**, *52*, 503–522. [[CrossRef](#)]
11. Oh, H.; Lee, S. On score vector-and residual-based CUSUM tests in ARMA–GARCH models. *Stat. Methods Appl.* **2018**, *27*, 385–406. [[CrossRef](#)]
12. Oh, H.; Lee, S. Modified residual CUSUM test for location-scale time series models with heteroscedasticity. *Ann. Inst. Stat. Math.* **2019**, *71*, 1059–1091. [[CrossRef](#)]
13. Lee, S. Location and scale-based CUSUM test with application to autoregressive models. *J. Stat. Comput. Simul.* **2020**, *90*, 2309–2328. [[CrossRef](#)]
14. Hwang, H.B.; Ang, H.T. A simple neural network for ARMA (p, q) time series. *Omega* **2001**, *29*, 319–333. [[CrossRef](#)]
15. Hossain, M.B.; Moon, J.; Chon, K.H. Estimation of ARMA model order via artificial neural network for modeling physiological systems. *IEEE Access* **2020**, *8*, 186813–186820. [[CrossRef](#)]
16. Zafar, T.; Kamal, K.; Mathavan, S. A Hybrid Approach for Noise Reduction in Acoustic Signal of Machining Process Using Neural Networks and ARMA Model. *Sensors* **2021**, *21*, 8023. [[CrossRef](#)]
17. Giacomazzo, M. Three Essays on Shrinkage Estimation and Model Selection of Linear and Nonlinear Time Series Models. Ph.D. Thesis, Arizona State University, Tempe, AZ, USA, 2018.
18. Harchaoui, Z.; Lévy-Leduc, C. Multiple change-point estimation with a total variation penalty. *J. Am. Stat. Assoc.* **2010**, *105*, 1480–1493. [[CrossRef](#)]
19. Chan, N.H.; Yau, C.Y.; Zhang, R.M. Group LASSO for structural break time series. *J. Am. Stat. Assoc.* **2014**, *109*, 590–599. [[CrossRef](#)]
20. Jin, B.; Shi, X.; Wu, Y. A novel and fast methodology for simultaneous multiple structural break estimation and variable selection for nonstationary time series models. *Stat. Comput.* **2013**, *23*, 221–231. [[CrossRef](#)]
21. Qian, J.; Su, L. Shrinkage estimation of common breaks in panel data models via adaptive group fused lasso. *J. Econom.* **2016**, *191*, 86–109. [[CrossRef](#)]
22. Lee, S.; Seo, M.H.; Shin, Y. The lasso for high dimensional regression with a possible change point. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2016**, *78*, 193. [[CrossRef](#)]
23. Jin, B.; Wu, Y.; Shi, X. Consistent two-stage multiple change-point detection in linear models. *Can. J. Stat.* **2016**, *44*, 161–179. [[CrossRef](#)]
24. Ciuperca, G.; Maciak, M. Change-point detection in a linear model by adaptive fused quantile method. *Scand. J. Stat.* **2020**, *47*, 425–463. [[CrossRef](#)]
25. Lee, S.; Lee, S.; Moon, M. Hybrid change point detection for time series via support vector regression and CUSUM method. *Appl. Soft Comput.* **2020**, *89*, 106101. [[CrossRef](#)]
26. Lee, S.; Lee, S. Change Point Test for the Conditional Mean of Time Series of Counts Based on Support Vector Regression. *Entropy* **2021**, *23*, 433. [[CrossRef](#)] [[PubMed](#)]
27. Looney, C.G. *Pattern Recognition Using Neural Networks: Theory and Algorithms for Engineers and Scientist*; Oxford University Press Inc.: Oxford, UK, 1997.
28. Swanson, N.R.; White, H. A model-selection approach to assessing the information in the term structure using linear models and artificial neural networks. *J. Bus. Econ. Stat.* **1995**, *13*, 265–275.
29. Hannan, E.J.; Rissanen, J. Recursive estimation of mixed autoregressive-moving average order. *Biometrika* **1982**, *69*, 81–94. [[CrossRef](#)]
30. Cobb, G.W. The problem of the Nile: Conditional solution to a changepoint problem. *Biometrika* **1978**, *65*, 243–251. [[CrossRef](#)]

31. MacNeill, I.B.; Tang, M.; Jandhyala, V.K. A search for the source of the Nile's change-points. *Environmetrics* **1991**, *2*, 341–375. [[CrossRef](#)]
32. Wu, W.B.; Zhao, Z. Inference of trends in time series. *J. R. Stat. Soc.* **2007**, *69*, 391–410. [[CrossRef](#)]
33. Li, W.K. *Diagnostic Checks in Time Series*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2003.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.