fractal and fractional

MDPI

Article

On Reservoir Computing Approach for Digital Image Encryption and Forecasting of Hyperchaotic Finance Model

Amr Elsonbaty [1,2,*], A. A. Elsadany [1,3] and Waleed Adel [2,4]

[1] Department of Mathematics, College of Science and Humanities in Al-Kharj, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[2] Mathematics and Engineering Physics Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt
[3] Basic Science Department, Faculty of Computers and Informatics, Suez Canal University, Ismailia 41522, Egypt
[4] Department of Technology of Informatics and Communications, Université Française d'Egypte, Ismailia Desert Road, El Shorouk, Cairo 11837, Egypt
* Correspondence: a.elsonbaty@psau.edu.sa

Abstract: Forecasting the dynamical behaviors of nonlinear systems over long time intervals represents a great challenge for scientists and has become a very active area of research. The employment of the well-known artificial recurrent neural networks (RNNs)-based models requires a high computational cost, and they usually maintain adequate accuracy for complicated dynamics over short intervals only. In this work, an efficient reservoir-computing (RC) approach is presented to predict the time evolution of the complicated dynamics of a fractional order hyperchaotic finance model. Compared with the well-known deep learning techniques, the suggested RC-based forecasting model is faster, more accurate for long-time prediction, and has a smaller execution time. Numerical schemes for fractional order systems are generally time-consuming. The second goal of the present study is to introduce a faster, more efficient, and simpler simulator to the fractional order chaotic/hyperchaotic systems. The RC model is utilized in a proposed RC-based digital image encryption scheme. Security analysis is carried out to verify the performance of the proposed encryption scheme against different types of statistical, KPA, brute-force, CCA, and differential attacks.

Keywords: reservoir-computing; hyperchaos; finance model; fractional order models; encryption

Citation: Elsonbaty, A.; Elsadany, A.A.; Adel, W. On Reservoir Computing Approach for Digital Image Encryption and Forecasting of Hyperchaotic Finance Model. *Fractal Fract.* **2023**, *7*, 282. https://doi.org/10.3390/fractalfract7040282

Academic Editor: Viorel-Puiu Paun

Received: 23 February 2023
Revised: 14 March 2023
Accepted: 21 March 2023
Published: 24 March 2023

1. Introduction

Significant advancements in communication systems, information processing, and data transmission mark the technological revolution of the twenty-first century. The amount of information being created, stored, or transmitted every second on earth is unparalleled. Today's communication systems differ in complexity, transfer rate, speed, and power consumption, based on their intended purposes and scopes. Nowadays, many tools can be utilized to transmit these data, and one of the most important among them is digital imaging. Digital images are used to convey a large amount of data and have many applications in various fields such as military, medical, and daily life [1,2]. When these data are transmitted over a wireless network through digital images, two important factors have to be taken into consideration. The first is to ensure that these data are securely transmitted over such an insecure wireless network. The other factor is to confirm the confidentiality of the data being transmitted over the network. To achieve this, image encryption is necessary in order to ensure this process. When transmitting multiple images at once, single-image encryption is no longer applicable; therefore, multiple-image encryption is considered to balance security and efficiency. In recent decades, image encryption algorithms have been widely used due to their excellent security properties. For example, Chuying et al. [3] employed a novel four-image encryption scheme based on several

approaches. Through these proposed techniques, they succeeded in significantly reducing the amount of key data, providing a more efficient encryption technique. In addition, Rafik et al. [4] proposed a privacy-preserving encryption cryptosystem that helps to protect patients' privacy as a direct application of digital image encryption for E-health systems. In addition, Ying et al. [5] developed an encryption scheme for optical image color, based on the two-dimensional nonlinear coupled map lattice. The application of this new approach results in a large key space and strong resistance to attacks, giving this scheme the advantage of providing more security to the transmitted digital images. Gao et al. [6] introduced a multiple-image encryption algorithm that utilizes single-image encryption through a chaotic system. Based on this, they achieved a good performance for the encryption process with excellent encryption speed. Finally, Elsadany et al. [7] designed a new hybrid technique for physical layer encryption and watermarking. The designed technique is based on discrete memristive chaos, and finally tests the proposed scheme for possible brute attacks, revealing good performance. With the importance of digital image encryption, many other techniques have been employed for this purpose, including artificial intelligence-based approaches that provide reliability and achieve great results.

During the last few years, artificial neural networks (ANNs) have attracted increasing interest from engineers, computer scientists, and mathematicians. Indeed, ANNs have successfully played an important role in simulating different real-life models due to their ability to provide a clear insight into their dynamics. Many branches of science and engineering benefit from this revolutionary concept, and have been used ever since in providing a greater understanding into the behaviors of these models. Object detection [8], natural language processing [9], autonomous vehicle driving [10], and security [11] are only some of the examples of the plethora of applications where the ANNs can be utilized. In a general form, the ANNs can be categorized into several types, based on the structure of the network. The first type is the feedforward neural network, which specifically depends on dividing the neurons into a specific number of layers, and then the processed signal flows forward in one direction. The convolutional neural network [12] or Shift Invariant is a more complicated ANNs that has been widely used in the field of object detection and face recognition. There are observed deficiencies in simulating temporal signals with the feedforward ANNs. Thus, another form of the ANNs, namely, the recurrent neural networks RNNs [13,14] have been developed. This type of ANNs depends on recruiting the neuron which helped in encoding the input signal into the internal state of the network itself. Consequently, it leads to a greater realization of the short-term memory concept. Indeed, the RNNs were considered to be an appropriate approach for forecasting temporal signals, but with the problem of having high computational costs. Moreover, some complex nonlinear behaviors are found to be difficult to be processed by RNNs.

The continuous efforts of researchers in this field aim at developing more accurate and capable time series forecasting techniques with more simple structures and less computational costs. This leads to the development of what is known as the reservoir-computing (RC) paradigm [15–22]. The RC technique has two ongoing aspects, named research and applications, which are being explored. The RC technique has proven its reliability and usefulness in many applications for its ability to provide more realistic simulations, avoiding the problems with the RNN. The main difference between the RC and RNNs approaches is that in the RC paradigm, the training process is required only for the weights that are connected to the output layer. In addition, the most important advantage of the RC technique over other conventional techniques is that the RC requires fewer weight parameters to be trained. Thus, the training process required for the RC is less than the other methods, especially the RNNs. In addition, the training process of the RC becomes linear and can be treated as linear regression. Thus, this robust, simple, and fast training algorithm can be used to simulate the given dynamical system.

The RC approach opens the door for many applications, as the concept of RC has helped with many problems from different disciplines, regardless of their complexity. In-

deed, the RC models can efficiently utilize the system's dynamics to achieve highly complex tasks with the involvement of some dependent data. Figure 1 gives a simple schematic diagram that shows the structure of the RC model. The RC models can be universally extended to a wider type of application compared to other ANNs approaches. Butcher et al. adapted the RC scheme for simulating the non-linear time series data [23]. Applications in wireless communications have been investigated with the aid of the RC algorithm in Jaeger et al. in [24], and have improved the accuracy of the predicted time series. In addition, Liu et al. [25] investigated the possible application of the RC frame in the field of information security through chaotic synchronization. Another application in weather forecasting for a short period was explained in [26] by Ferreira et al., revealing an excellent prediction of the hourly wind speed for a short period using RC prediction. Escalona-Morán et al., in [27], adapted the RC technique for classifying the electrocardiogram with logistic regression. In addition, the application of the RC algorithm has been extended to the processing of noisy image recognition and simulation, as indicated in [28] by Jalalvand et al. It has been found that the RC technique has many other interesting applications such as controlling robotic systems and leading, which has led to the development of the drone industry. Lukoševičius et al., in [17], illustrates the trends in the robotic field through the application of the RC technique. The RC paradigm is observed to have a superiority over other relatively similar techniques in performing many tasks. The efficiency of the RC can be maximized through the careful choice of a suitable reservoir with an adequate capacity for storing information. Interestingly, the RC scheme has shown to be very helpful and has achieved some brilliant performances in the area of computer science. Speech recognition is one of these related topics that have various benefits from the RC technique, as presented by Verstraeten in [29]. Also, some new materials and devices were designed with the aid of the RC technique, including some photonic modules, as provided by Martinenghi et al. in [30], Vandoorne et al. in [31], and Antonik et al. in [32]. Memristors-based circuits are also one of the crucial devices that employ the RC technique in hardware implementation [33–35], by taking into account the advantages of their analogous resistive switching properties [36–39].

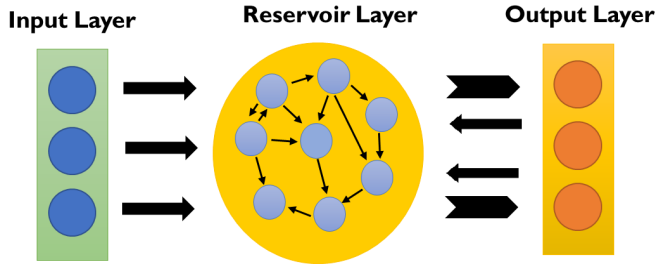


Figure 1. Schematic diagram of the structure of reservoir computer.

In the field of forecasting models for financial/stock temporal data trends, RC has been also used for these simulations. For example, Wang et al. [40] developed a new approach based on the RC technique for predicting the livestock market index behavior based on small-world topologies. The resulting simulations are shown to be competitive compared to other deep-learning models. An RC computing approach was developed in [41] for analyzing integer-order time-delay-controlled financial systems. Another model of computing the strategies for monthly time series prediction was introduced by Wyffels in [42], which may help companies to arrange their investment and production needs. Other models include estimating risk bounds [43] and visualization methods for temporal data [44], etc.

Historically, the hyperchaos term was first introduced by Rossler, who derived a new four-dimensional hyperchaotic system named after him. These hyperchaotic systems are characterized by the fact that the corresponding strange attractors have more than one positive Lyapunov exponent. This implies that the hyperchaotic system exhibits more complicated chaotic dynamical behaviors relative to the conventional chaotic systems.

During the last few years, more attention has been given to these hyperchaotic systems with their wide range of applications in engineering and science. In addition, the applications of chaotic and hyperchaotic systems in finance and economy better simulate some practical and theoretical aspects that are related to this field. For example, Szumiński et al. in [45] investigated two new types of hyperchaotic systems and revealed some of their dynamics, with possible applications in finance through bifurcation diagrams and the Lyapunov exponents spectrum. Yu et al. in [46] proposed a new hyperchaotic finance system that is a modification to the chaotic finance system, by adding some new state variables. A new adaptive synchronization scheme for hyperchaotic finance systems with unknown parameters and bounded disturbances was introduced by Vargas et al. in [47]. Cao et al. [48] created a new four-dimensional hyperchaotic model and studied several control parameters for this system. These are only a few of the works of chaotic systems in finance. For more details regarding the hyperchaotic systems and their applications in finance, the reader may refer to [49–57]. All of the previously mentioned models benefit from understanding their dynamics, with the discovery of the fractional calculus that brings power to their perspective.

Fractional calculus is a branch of mathematics that employs non-local operators in the form of derivatives and integrals of arbitrary orders. It was first introduced by Leibniz back in the 17th century, and has since been used to accurately model a wide variety of phenomena, from fluid dynamics [57] to financial markets [58], with high precision. Indeed, fractional calculus has been used to model many systems from different fields, including physics, engineering, economics, and biology. There have been many definitions of fractional operators and each of these definitions has its merit. One of the most widely used fractional operators is the Caputo fractional derivative operator. It has been applied to many areas such as viscoelasticity [59], wave propagation [60], and diffusion processes [61]. The Caputo fractional derivative has also been used in the study of chaotic systems and fractals [62]. The key advantage of the Caputo fractional derivative is that it allows for more flexibility than other types of fractional order derivatives. Thus, using this fractional operator in simulating different models gives an added great understanding of their behavior.

The motivation of the work can be summarized as follows: Artificial recurrent neural networks (RNNs) can be trained and utilized to perform complex tasks such as the forecasting of complicated nonlinear time series, handwriting recognition, and language processing. However, there are two common issues associated with RNNs. The first issue is that they require training data of massive size, and the training process itself is challenging and computationally expensive. Moreover, the corresponding hardware realization is not simple and requires expensive elements. The second and more critical issue is that the accuracy of RNNs prediction is maintained basically for short time intervals, while it can be lost for long-time forecasting. The first goal of the present work is to introduce an RC-based model for the highly accurate long-time forecasting of complicated hyperchaotic financial time series, for the first time. We verified the superiority of the suggested RC model in terms of accuracy, long-time prediction, simplicity, and running time.

On the other hand, the study and analysis of fractional order systems (FOS) have become a very active area of research over the last two decades. Numerical schemes for FOS are generally time-consuming, and the number of required arithmetic operations rapidly increases with the increment of solution intervals. The second goal of the present study is to introduce a faster, more efficient, and simpler simulator to the fractional order chaotic/hyperchaotic systems. The presented scheme can emulate fractional order dynamical systems and accurately predict the solution time series in a fast time, compared with conventional numerical schemes.

The following fractional hyperchaotic finance system of order σ is considered

$$\begin{array}{ll} D^{\sigma}u(t) & = w(t) + (v(t) - \alpha)u(t) + w(t), \\ D^{\sigma}v(t) & = 1 - \beta v(t) - u(t)^2, \\ D^{\sigma}w(t) & = -u(t) - \gamma w(t), \\ D^{\sigma}z(t) & = -\mu u(t)v(t) - \kappa w(t). \end{array} \tag{1}$$

The above model is an extension of the integer order model presented in [46] by incorporating the Caputo fractional order derivative. The four state variables in the model are the interest rate u, the demand on the investment v, the exponent of price w, and the profit margin average z. In addition, the parameters presented in the model (1) can be defined, as α denotes the saving parameter, β refers to the investment cost, γ is the elasticity of demand on commercials, and finally, μ and κ are constants with positive values. System (1) exhibits hyperchaotic behavior when the parameters take the values of $\alpha = 0.9, \beta = 0.2, \gamma = 1.5, \mu = 0.2$, and $\kappa = 0.17$. The initial values of the model (1) are taken as $(u, v, w, z) = (1, 2, 0.5, 0.5)$ in our simulations. It will be shown that the RC model reduces the execution time of the forecasting process from around 59 s in the well-known Long Short-Term Memory (LSTM)-type RNNs [63,64], and 5.1 s in the PECE numerical method, to 0.85 s.

Finally, the image encryption schemes which rely on the hyperchaotic fractional order systems inherit the high computational costs of the seed fractional chaos generators. The present work is an attempt to utilize the aforementioned benefits of the RC machine learning approach in a proposed image encryption scheme. The proposed scheme is much faster, and the total processing time of the encryption algorithm is greatly reduced from 6.2 s (when conventional fractional chaotic systems are used) to 1.85 s. In addition, security analysis reveals that the presented encryption technique is resistant to brute force, statistical, differential, KPA, and CCA attacks.

The organization of the paper is summarized as follows. The mathematical model for the RC technique is presented in Section 2. Numerical simulations are conducted, and the performance of the scheme is evaluated in Section 3. In addition, comparisons with the LSTM technique are performed. The proposed RC-based encryption scheme and the associated security analysis are presented in Sections 4 and 5, respectively. The conclusion and suggested future work are presented in Section 6.

2. RC Model for the Forecasting of the Hyperchaotic Financial Time Series

In this section, we introduce the main procedure of the RC approach. First, suppose that $\chi(n)$ is defined as an N_1 dimensional input signal at discrete time n. In addition, we define $\Xi(n)$ and $\Psi(n)$ to be the N_2 dimensional vectors of the reservoir activation and the updates of the RC state, respectively. The updated equations of the RC system can take the form

$$\Psi(n) = (1 - \varrho)\Xi(n) + \varrho\eta(n), \tag{2}$$

$$\Xi(n) = \Re(\Phi^{in}[1; \chi(n)] + \Phi\Xi(n-1)), \tag{3}$$

where the nonlinear activation function $\Re(.)$ is usually chosen in the form of $\tanh(.)$, $\Phi^{in} \in \mathbb{R}^{N_2 \times (N_1+1)}$ is the input weight matrix, $\Phi \in \mathbb{R}^{N_1 \times N_2}$ is the recurrent weight matrix for the hidden layer, $\varrho \in [0, 1]$ is the rate of leakage, and $[.,.]$ is the concatenation of column vectors. For the case where $\varrho = 1$, it follows that $\Psi(n) = \eta(n)$.

The expression of the linear readout output layer $\Upsilon^{out}(n) \in \mathbb{R}^{N_{out}}$ can take the following form

$$\Upsilon^{out}(n) = \Phi^{out}[1; \chi(n); \Psi(n)],$$

where $\Phi^{out} \in \mathbb{R}^{N_{out} \times (1+N_1+N_2)}$ is the matrix of the output weights. One may note that in some special cases of the RC designs, there are some extra feedback output layers of the function $\Re$ in Equation (2), such that

$$\Xi(n) = \Re(\Phi^{in}[1; \chi(n)] + \Phi\Xi(n-1) + \Phi^{fb}\Upsilon^{out}(n-1)),$$

where Φ^{fb} is defined as the output feedback matrix.

The training process of the RC supervised machine learning paradigm uses the input signal $\chi(n)$ and updates output weights such as $\Upsilon^{out}(n) \in \mathbb{R}^{N_{out}}$, which matches the specified target sequence $\Upsilon^{TR}(n) \in \mathbb{R}^{N_{out}}$. In particular, this goal is achieved by minimizing the root mean square error (RMSE) $\mathbb{E}$, which takes the following form

$$\mathbb{E} = \frac{1}{N_{out}} \sum_{i=1}^{N_{out}} \sqrt{\frac{1}{m} \sum_{n=1}^{m} (\Upsilon_i^{out}(n) - \Upsilon_i^{TR}(n))^2},$$

where m denotes the training sequence length. When the training process is accomplished successfully, the RC can be utilized to predict the future states of the sequence.

To operate with the RC, we can summarize the main steps for the RC forecasting procedure, as follows:

Step I

Create the reservoir of the RC with random values of internal weights. The associated matrices Φ^{in} and Φ are preferred to be sparse matrices, in order to simplify the calculations and to reduce the running time.

Step II

The training vector $\chi(n)$ is applied, and the resulting activation state vector $\Psi(n)$ is acquired. The harmful effects of the outliers of the input data can be suppressed by employing the normalization of the input data.

Step III

The output weight matrix Φ^{out}, which minimizes RMSE $\mathbb{E}$ between Υ^{out} and Υ^{TR}, is evaluated.

Step IV

The final step is to use the trained matrix Φ^{out} to forecast the output of new input data. Generally, the size of the reservoir should exceed the dimension of the input data vector, multiplied by the number of time steps of the input data that are required to be memorized by the RC to complete the assigned task. In addition, the sparse matrix Φ must be initially scaled to an acceptable value of spectral radius that preserves the fading memory characteristics of the RC. More training and tuning procedures should be performed to scale the matrix Φ again to ensure the achievement of the suitable form. The appropriate values for ϱ are determined by the time scales of the training signal and the target signal.

Now, in the following subsection, we illustrate the training procedure of the RC.

Procedure of Training the RC

In this subsection, we shall illustrate the main procedure steps for training the RC. First, the output of the RC over a defined period $s = 1, 2, ..., m$ can be defined as

$$\Lambda = \Phi^{out}\mathcal{C},$$

where $\Lambda = [\Upsilon^{out}(1)\Upsilon^{out}(2)...\Upsilon^{out}(m)]$ and $\mathcal{C}$ is the concatenation of $[1; \chi(n); \Psi(n)]$. Then, the output matrix Φ^{out} in optimal form is attained by minimizing $\mathbb{E}$ between the teacher data and the RC output data, or by solving the following equation

$$\Lambda^{TR} = \Phi^{out}\mathcal{C}.$$

Indeed, we need to solve an overdetermined system of equations in order to achieve this task. Thus, the Tikhonov regularization technique (ridge regression) can be employed as a useful and efficient tool to achieve this goal. The objective function obeying this approach can be put in the following form

$$\Phi^{out} = \underset{\Phi^{out}}{argmin} \frac{1}{N_{out}} \sum_{i=1}^{N_{out}} \Big(\sum_{n=1}^{p} (\Lambda_i(n) - \Lambda_i^{TR}(n))^2 + \varpi \big\| \Phi_i^{out} \big\|^2 \Big),$$

where the norm of Φ_i^{out} is the well-known Euclidean norm of the ith row of the matrix Φ^{out}. This regularization reduces the effect of overfitting and feedback instability. Finally, the optimal form of the matrix Φ^{out} can be written as

$$\Phi^{out} = \Lambda^{TR} \mathcal{C}^T (\varpi \mathcal{I} + \mathcal{C}\mathcal{C}^T)^{-1},$$

where ϖ is the regularization parameter and $\mathcal{I}$ denotes the identity matrix.

The next section is devoted to illustrating and discussing the obtained results of the RC paradigm.

3. Numerical Simulations

In this section, we use the suggested RC forecasting approach to estimate the complicated time series solution of the hyperchaotic finance model (1). The phase portraits of the fractional order hyperchaotic finance model (1), obtained at the different values of σ are shown in Figure 2. The processed time series length equals $10,000$ for each state variable of the model. The training part is constructed from the first 7000 elements of the data vector. The remaining 3000 elements constitute the test validation set. In addition, the number of RC nodes is set at 32. Three cases for fractional order σ are considered. Figures 3–6 depict the training and prediction results at $\sigma = 1$ for different state variables of the fractional order hyperchaotic finance model (1). Figures 7–10 illustrate the training and prediction results at $\sigma = 0.95$ for the state variables of the model (1). Moreover, Figures 11–14 illustrate the training and prediction results at $\sigma = 0.8$ for the state variables of the model (1).

Long Short-Term Memory (LSTM) [63,64] is a deep-learning technique that can be regarded as a special class for recurrent neural networks. The LSTM networks have the advantages of employing additional gates and better controlling the processed information in hidden cells. So, LSTM networks can learn long-term relationships in time series and make the classification of sequential data more effective than the classical recurrent neural networks.

The LSTM technique is applied for the forecasting process of our financial time series. The length of the training part of the data vector is set at 8000, and the size of the validation test part is 2000. Examples of the obtained results are shown in Figures 15 and 16. It is obvious that the results of the RC-based prediction process are more accurate over long intervals and show very small forecasting errors compared with the LSTM approach. In addition, the RC-based model is easily realizable and highly efficient, from the viewpoint of computational cost and execution time. Table 1 illustrates the comparisons between the two approaches, in terms of RMSE and execution time (in a s) for different cases of σ and different state variables.

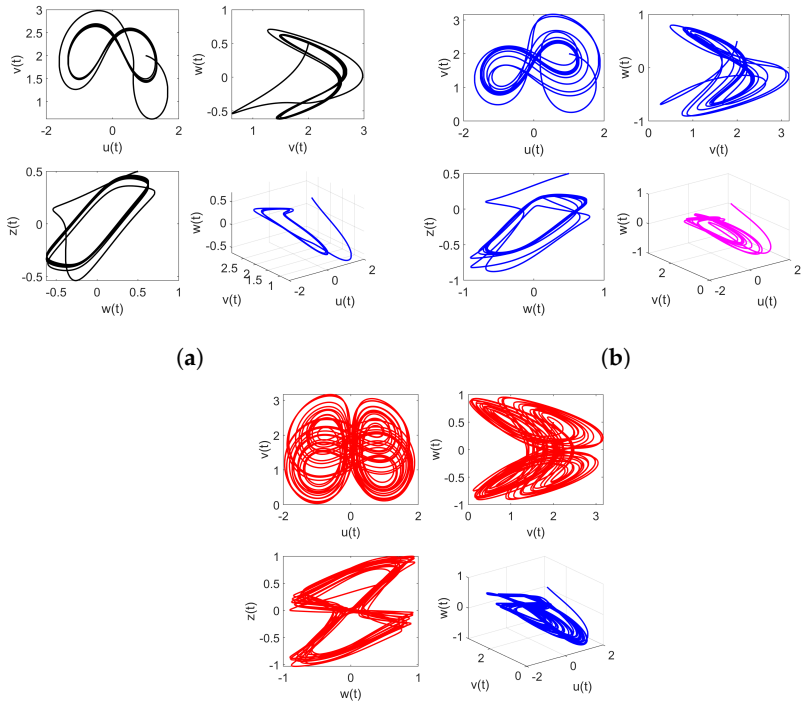


Figure 2. The phase portraits of the fractional order hyperchaotic finance model (1), obtained at the different values of σ: (**a**) $\sigma = 0.8$, (**b**) $\sigma = 0.95$, and (**c**) $\sigma = 1$.

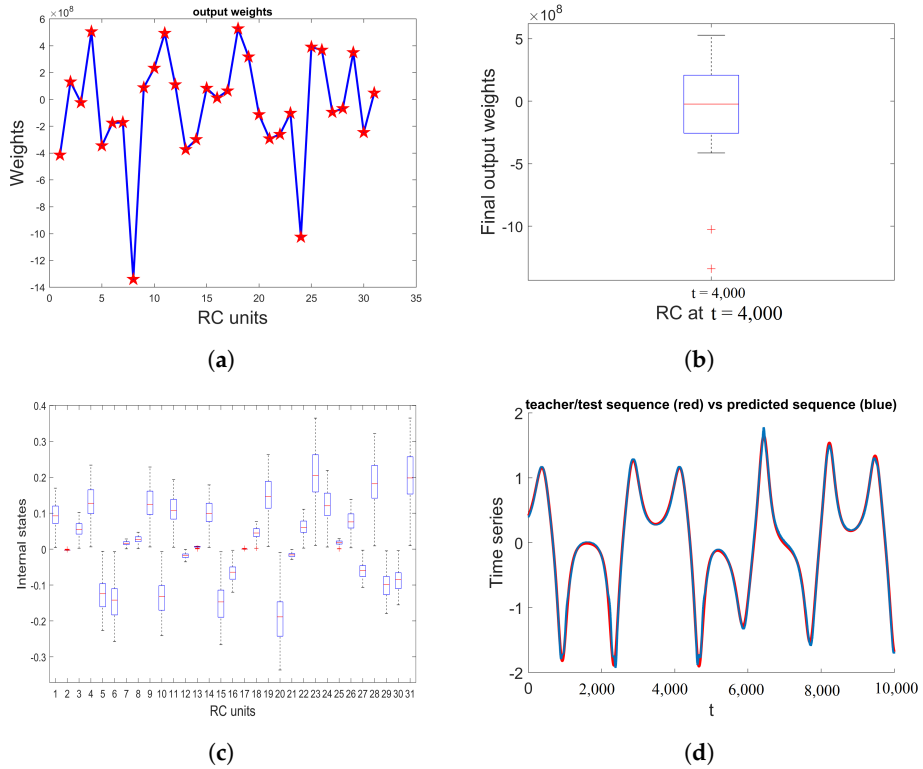
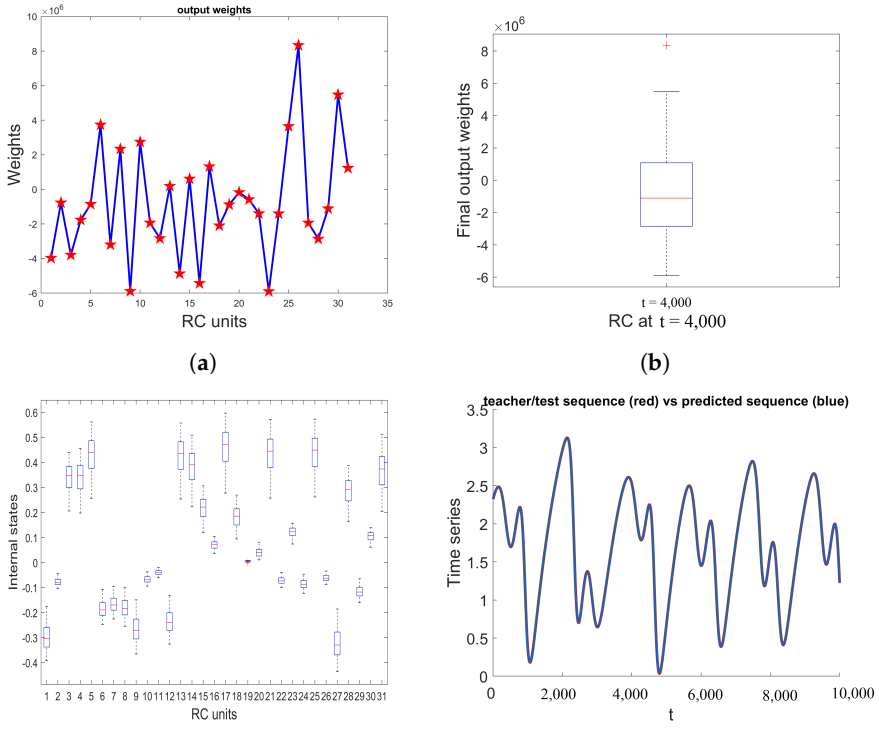


Figure 3. Training and prediction results of the RC model for $u(t)$ state variable of the model (1) at $\sigma = 1$. The 70% of input data are used as a teacher sequence, whereas the last 30% of the data are set as the validation test data. (**a**) The weights of RC nodes after the completion of training. (**b**,**c**) The spread of the weights of RC units, illustrated by using a box-and-whisker plot. (**d**) The training and test data in red color vs. predicted data in blue color.

Table 1. The performance of RC forecasting scheme and LSTM approach.

Predicted Series	RMSE (Suggested RC)	RMSE (LSTM)	Execution Time (RC)	Execution Time (LSTM)
u(t), $\sigma = 1$	0.010302	2.1758	0.8–0.9	55–63
v(t), $\sigma = 1$	0.015974	1.7028	0.8–0.9	55–63
w(t), $\sigma = 1$	0.022054	1.8142	0.8–0.9	55–63
z(t), $\sigma = 1$	0.018673	1.9533	0.8–0.9	55–63
u(t), $\sigma = 0.95$	0.0007084	1.5214	0.8–0.9	55–63
v(t), $\sigma = 0.95$	0.00068142	1.5003	0.8–0.9	55–63
w(t), $\sigma = 0.95$	0.00065223	1.6287	0.8–0.9	55–63
z(t), $\sigma = 0.95$	0.0095651	1.4169	0.8–0.9	55–63
u(t), $\sigma = 0.8$	0.0002302	1.2514	0.8–0.9	55–63
v(t), $\sigma = 0.8$	0.00079411	1.2210	0.8–0.9	55–63
w(t), $\sigma = 0.8$	0.00032459	1.3018	0.8–0.9	55–63
z(t), $\sigma = 0.8$	0.00022052	1.1074	0.8–0.9	55–63



(**a**) (**b**) (**c**) (**d**)

Figure 4. Training and prediction results of the RC model for $v(t)$ state variable of the model (1) at $\sigma = 1$. The 70% of input data are used as the teacher sequence, whereas the last 30% of the data are set as the validation test data. (**a**) The weights of RC nodes after the completion of training. (**b**,**c**) The spread of the weights of RC units, illustrated using a box-and-whisker plot. (**d**) The training and test data in red color vs. predicted data in blue color.

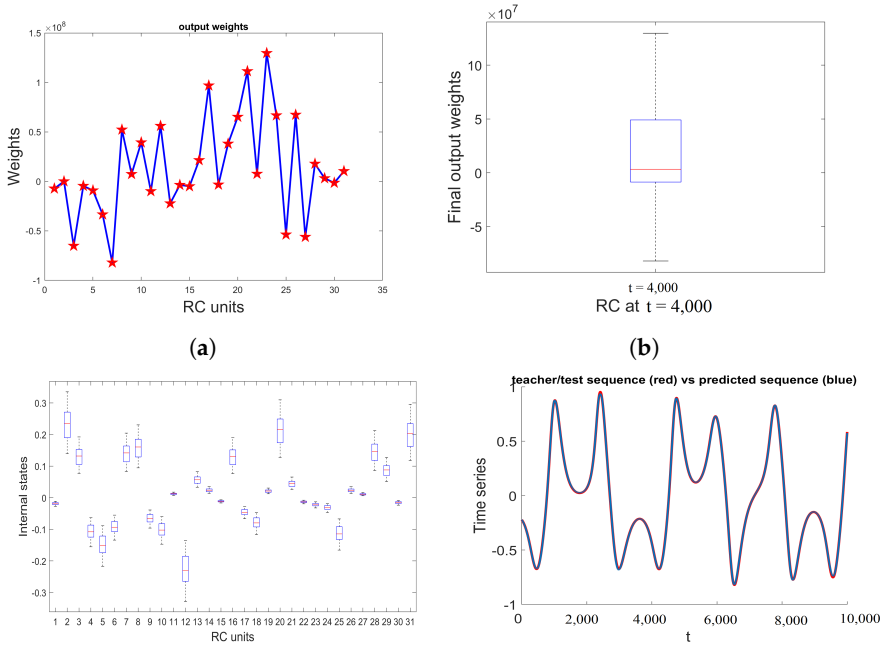


Figure 5. Training and prediction results of the RC model for $w(t)$ state variable of the model (1) at $\sigma = 1$. The 70% of input data are used as teacher sequence, whereas the last 30% of the data are set as validation test data. (**a**) The weights of RC nodes after the completion of training. (**b**,**c**) The spread of the weights of RC units, illustrated using box-and-whisker plot. (**d**) The training and test data in red color vs. predicted data in blue color.

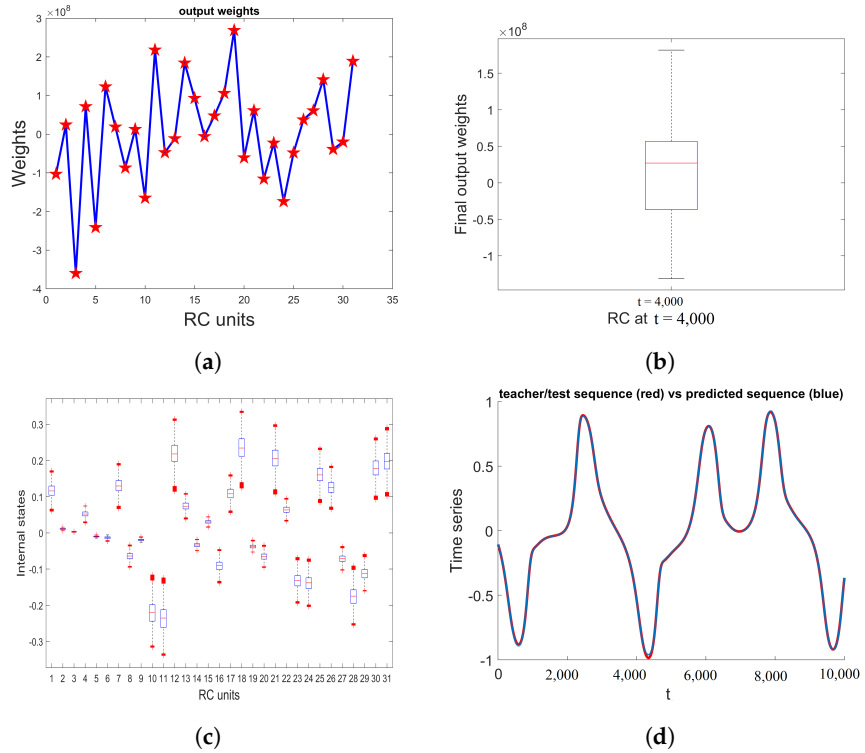
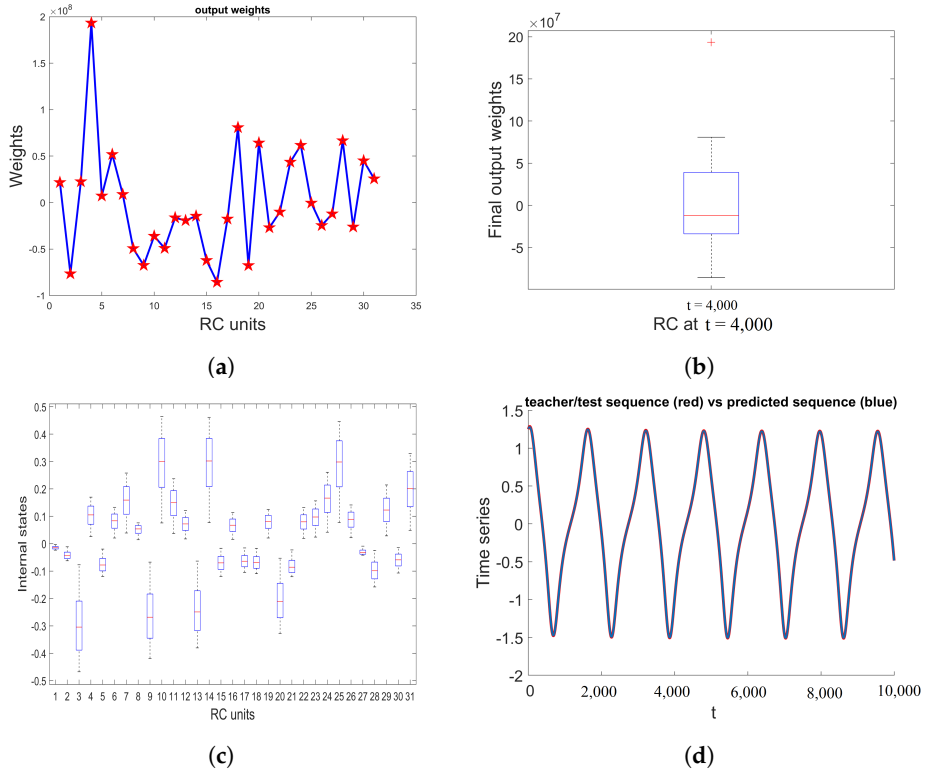


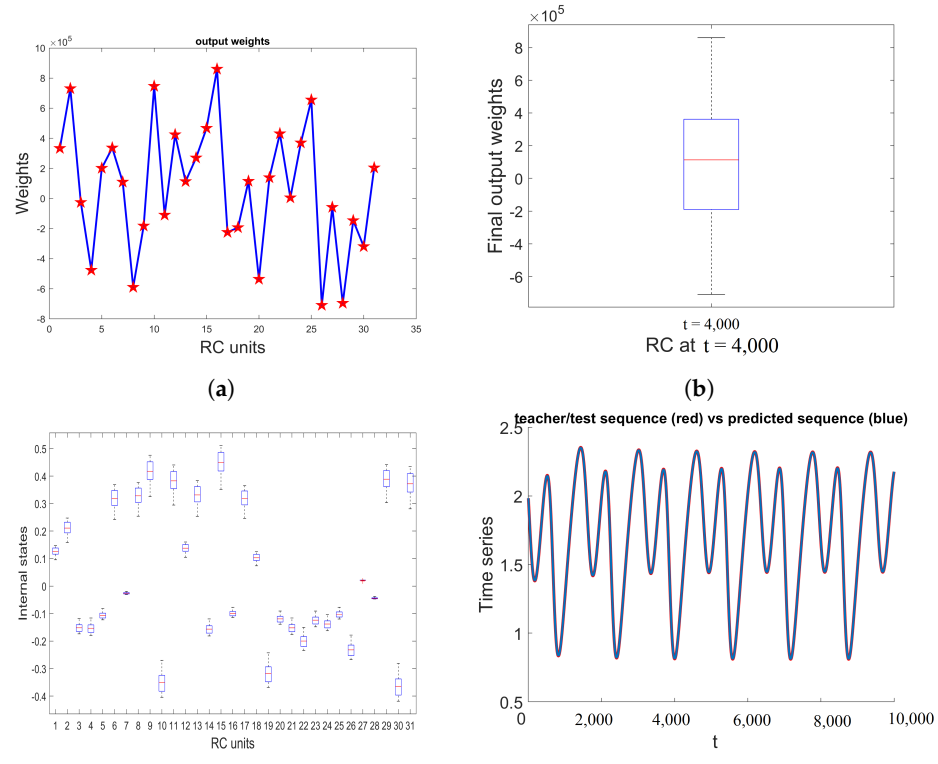
Figure 6. Training and prediction results of the RC model for $z(t)$ state variable of the model (1) at $\sigma = 1$. The 70% of input data are used as teacher sequence, whereas the last 30% of the data are set as validation test data. (**a**) The weights of RC nodes after the completion of training. (**b**,**c**) The spread of the weights of RC units illustrated using box-and-whisker plot. (**d**) The training and test data in red color vs. predicted data in blue color.



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

(**c**) (**d**)

Figure 7. Similar to Figure 3 but for the case where $\sigma = 0.95$.



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

(**c**) (**d**)

Figure 8. Similar to Figure 4 but for the case where $\sigma = 0.95$.

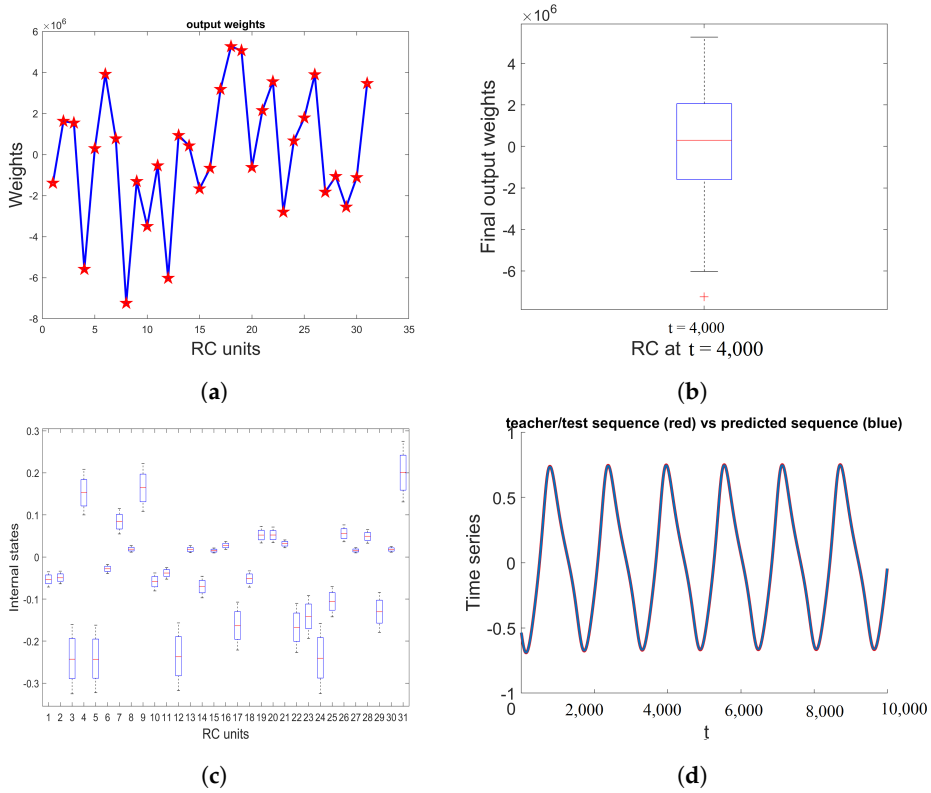


Figure 9. Similar to Figure 5 but for the case where $\sigma = 0.95$.

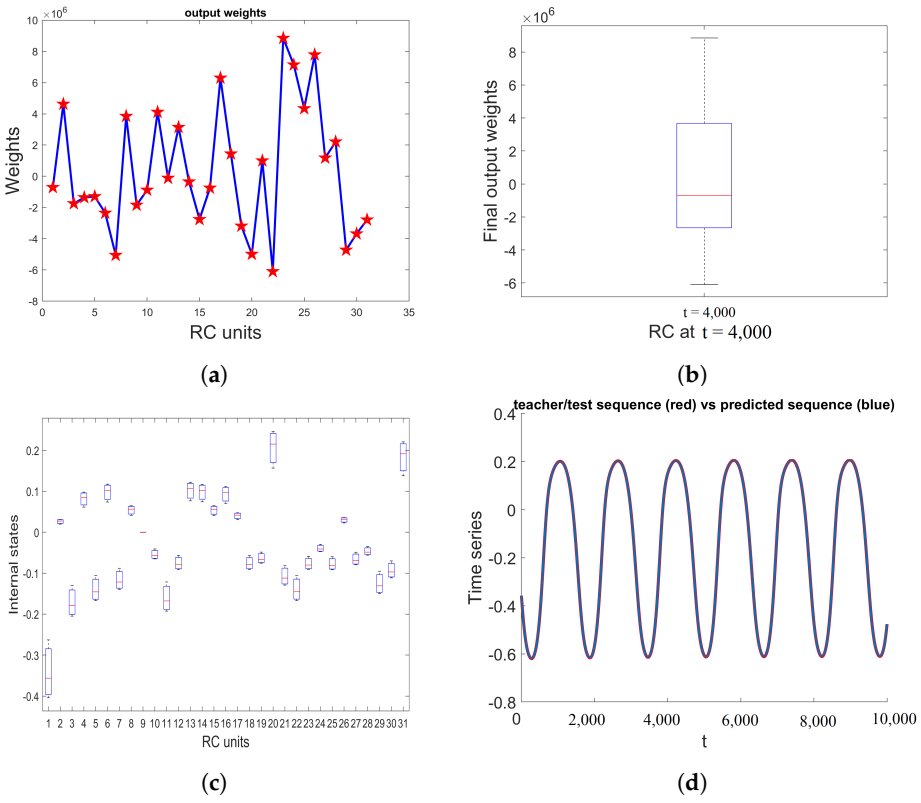


Figure 10. Similar to Figure 6 but for the case where $\sigma = 0.95$.

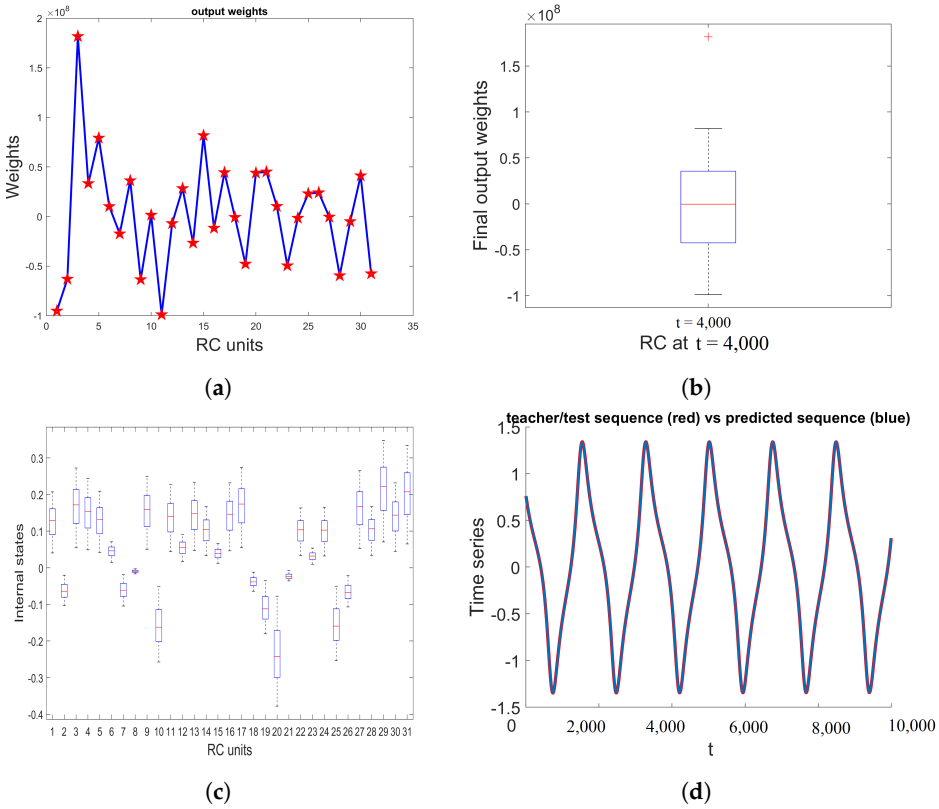


Figure 11. Similar to Figure 3 but for the case where $\sigma = 0.8$.

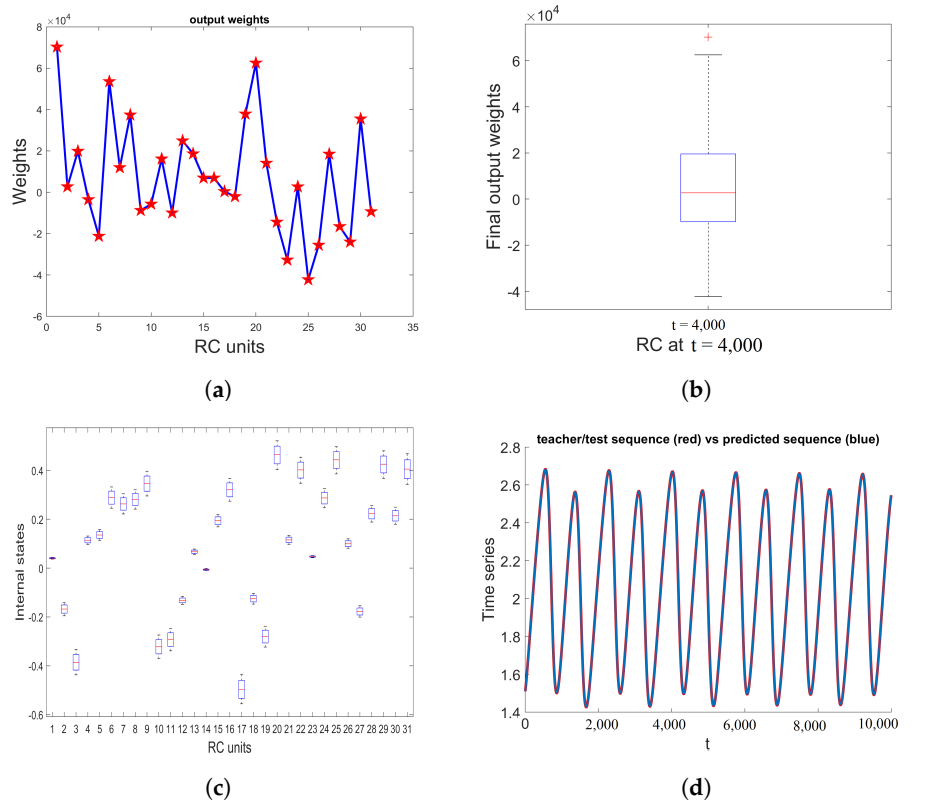


Figure 12. Similar to Figure 4 but for the case where $\sigma = 0.8$.

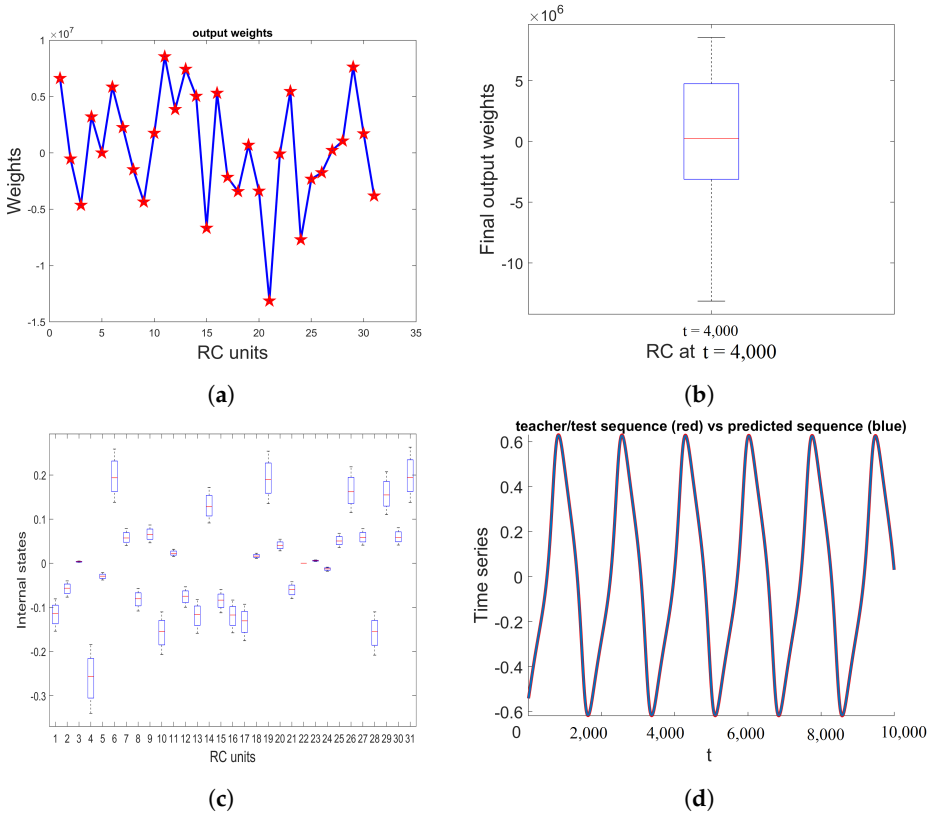


Figure 13. Similar to Figure 5 but for the case where $\sigma = 0.8$.

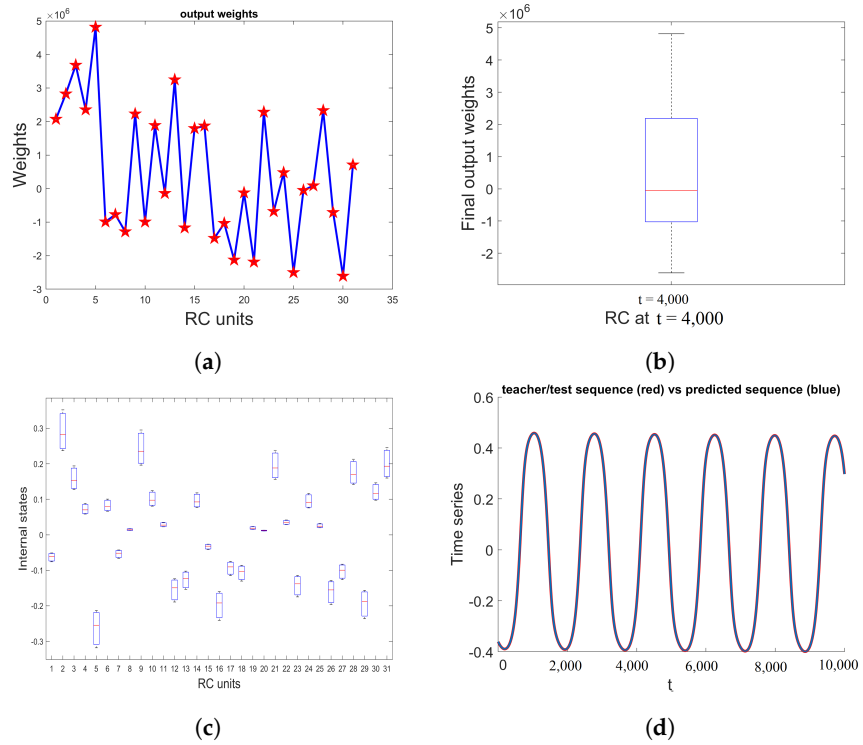


Figure 14. Similar to Figure 6 but for the case where $\sigma = 0.8$.

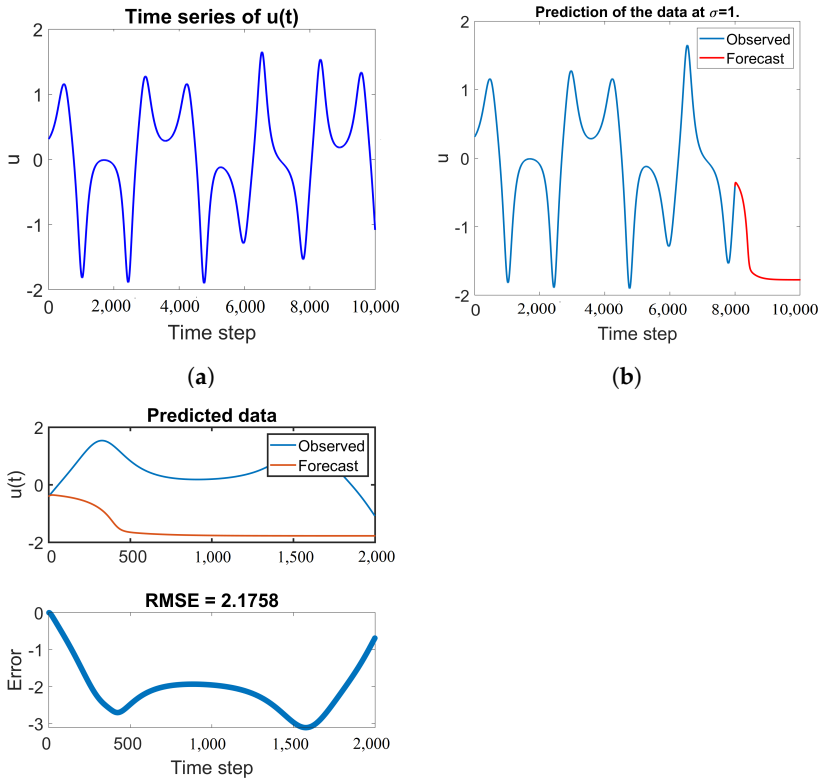


Figure 15. (**a**) The time series of $u(t)$ of the model (1) obtained at $\sigma = 1$. (**b**) The prediction of time series using the LSTM method. (**c**) The predicted data and the error between observed and predicted time series.

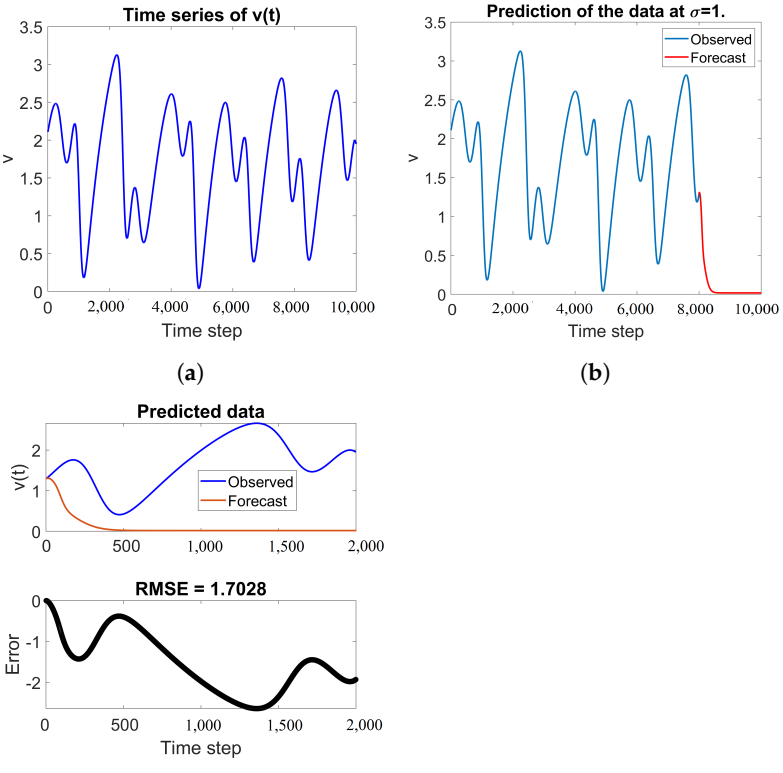


Figure 16. (**a**) The time series of $v(t)$ of the model (1) obtained at $\sigma = 1$. (**b**) The prediction of time series using the LSTM method. (**c**) The predicted data and the error between observed and predicted time series.

4. The Proposed RC-Based Image Encryption Scheme

In this section, the reservoir-computing model is employed as a core for the proposed new image encryption algorithm. More specifically, the RC paradigm is efficiently used to produce random-like time series for the image encryption process. In other words, the RC model is utilized as a simulator for a discrete-time fractional order hyperchaotic map. The generated time series can mimic the chaotic attractor of the fractional order hyperchaotic system in the present study. Indeed, the presented scheme is capable of simulating any selected fractional order hyperchaotic system. The advantages of this approach for encryption applications are (1) Low computational cost and distinct small running time. It is known that the numerical schemes for fractional order differential equations are time-consuming and require quite a high computational cost. (2) A larger space of secret parameters compared with conventional chaos-based encryption techniques.

Given an $M \times N$ colored plain image, the key steps for the proposed RC encryption algorithm are described as follows:

Key Steps for the RC Encryption Algorithm

Step 1. Define k_1 and k_2 by

$$k_1 = floor[\frac{M \times N}{a}],$$

$$k_2 = floor[\frac{M \times N}{b}],$$

where a and b are two preselected positive integers. Without loss of generality, they are set to $a = 50, b = 50$ in our simulations.

Step 2. Let $P_r(i, j)$ denote the pixel value of the red color component at position (i, j), then reshape the associated $M \times N$ matrix P_r into the row vector V_1^r by combining all rows of the matrix into a single row of size $1 \times M.N$. Similarly, reshape the matrix P_r into a single column V_2^r of size $M.N \times 1$ via the concatenation of its columns.

Step 3. Similar to Step 2, the green and blue color channels of the plain colored image are represented by the row vectors V_1^g and V_1^b, respectively. In addition, the column vectors of green and blue channels will be denoted by V_2^g and V_2^b, respectively.

Step 4. Create six plain-image-dependent and time-varying perturbation values $\vartheta_{1,2}^{r,g,b}$ by evaluating

$$\vartheta_{1,1}^{r,g,b} = \gamma\tau(t) + \frac{1}{\alpha(M \times N)^2} \sum_{i=1}^{k_1} V_1^{r,g,b}(i, j), \tag{4}$$

$$\vartheta_{2,1}^{r,g,b} = \gamma\tau(t) + \frac{1}{\alpha(M \times N)^2} \sum_{i=1}^{k_2} V_2^{r,g,b}(i, j), \tag{5}$$

$$\vartheta_{1,m}^{r,g,b} = \gamma\tau(t) + \frac{1}{\alpha(M \times N)^2} \sum_{i=(m-1)k_1+1}^{mk_1} V_1^{r,g,b}(i, j), \tag{6}$$

$$\vartheta_{2,n}^{r,g,b} = \gamma\tau(t) + \frac{1}{\alpha(M \times N)^2} \sum_{i=(n-1)k_2+1}^{nk_2} V_2^{r,g,b}(i, j), \tag{7}$$

where $m = 2, 3, ..., a$ and $n = 2, 3, ..., b$. Here, α and γ are two scaling factors. A secret baseline past time moment is selected, e.g., 07:53:19:579 11 March 2005, and the value of $\tau(t)$ is obtained by calculating the time difference up to the present time for when a plain image is encrypted. The unit of time is milliseconds or smaller units. The scaling factor γ is used to set the value of $\nu\tau(t)$ in the predetermined required range.

Step 5. The training data χ for RC has $3(a+b)$ elements. The six time series $\vartheta_{1,2}^{r,g,b}$ are utilized to perturb the training data, such that χ is updated by:

$$\chi = \chi + \vartheta_{1,2}^{r,g,b}. \tag{8}$$

Note that there are 6! possible choices for perturbing χ by the $\vartheta_{1,2}^{r,g,b}$ time series.

Step 6. The training process for the RC is carried out using updated χ, and the attained Y_i RC output is stored into a vector ρ of length $6M.N$.

Step 7. Define

$$\rho_r = mod[10^{10} \times \rho(j), M.N] + 1, j = 1, 2, ...M.N$$

$$\rho_g = mod[10^{10} \times \rho(j), M.N] + 1 + 1, j = M.N + 1, M.N + 2, ...2M.N$$

$$\rho_b = mod[10^{10} \times \rho(j), M.N] + 1, j = 2M.N + 1, 2M.N + 2, ...3M.N$$

where the repeated values are replaced with non-repeated ones in each vector in such a way where each vector covers all values in the range from 1 to $M.N$.

Step 8. The shuffling process for V_1^r, V_1^g, and V_1^b is carried by rearranging them as follows:

$$V_1^r(j) = V_1^r(\rho_r(j)),$$
$$V_1^g(j) = V_1^g(\rho_g(j)),$$
$$V_1^b(j) = V_1^b(\rho_b(j))$$

Step 9. The remaining $3M.N$ elements in the chaotic output of the RC are modified as follows

$$\varphi_r = mod[10^{10} \times \rho(j), M.N] + 1, j = 3M.N + 1, 3M.N + 2, ...4M.N$$

$$\varphi_g = mod[10^{10} \times \rho(j), M.N] + 1 + 1, j = 4M.N + 1, 4M.N + 1, ...5M.N$$

$$\varphi_b = mod[10^{10} \times \rho(j), M.N] + 1, j = 5M.N + 1, 5M.N + 2, ...6M.N$$

Step 10. Apply the bitwise XOR operation between the shuffled pixels and the modified chaotic sequences as follows

$$CV_1^r(j) = \varphi_r(j) \oplus V_1^r(j),$$
$$CV_1^g(j) = \varphi_g(j) \oplus V_1^g(j),$$
$$CV_1^b(j) = \varphi_b(j) \oplus V_1^b(j). \tag{9}$$

Step 11. The resulting cipher image is finally attained by reshaping $CV_1^{r,g,b}$ into the $M \times N$ cipher image.

Step 12. The decryption process is executed by reversing the aforementioned steps.

Three colored images are used in the numerical simulations, namely, Baboon, Pepper, and House images, respectively. Figure 17 shows the plain, shuffled, cipher images for the three images when the proposed algorithm is applied.

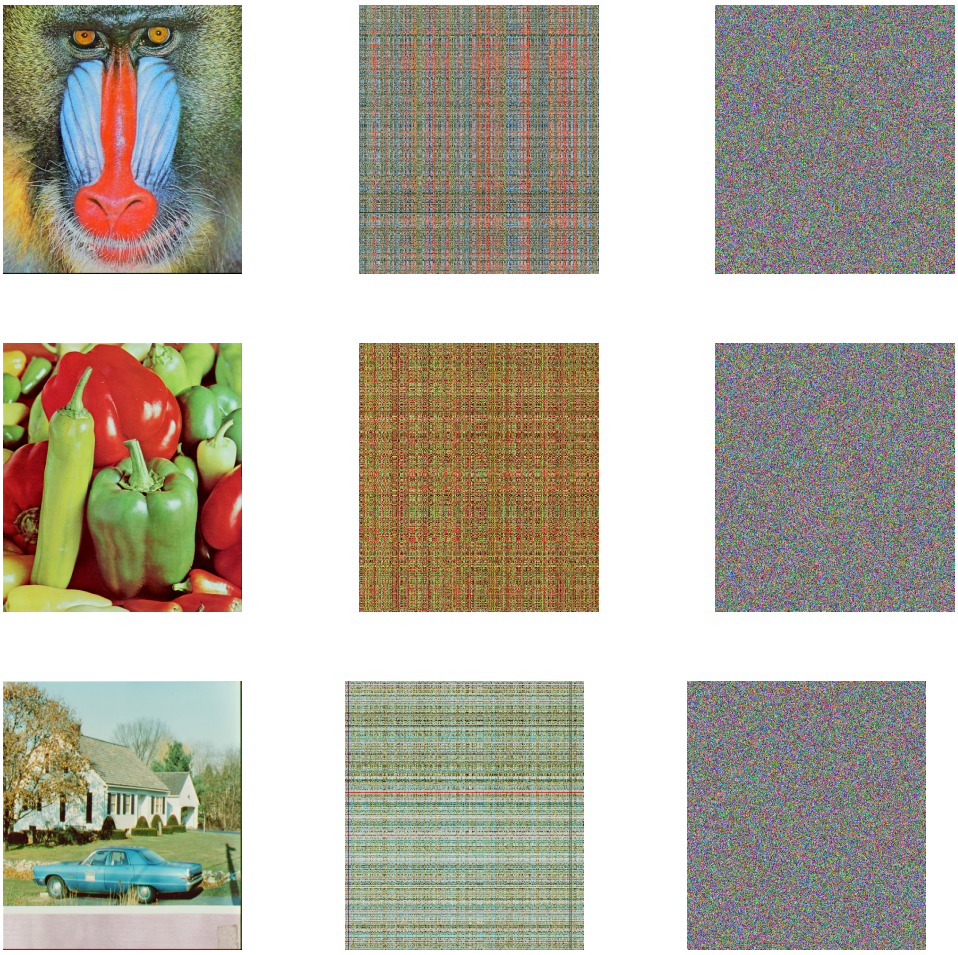


Figure 17. The plain, shuffled, and encrypted images for Baboon, Pepper, and House images.

5. Security Analysis of the RC Encryption Scheme

Now, the performance of the proposed RC-based digital image encryption technique is evaluated to verify its reliability against possible attacks such as brute force attacks, differential attacks, statistical attacks, chosen-plaintext attacks (CPA), and chosen-ciphertext attacks (CCA).

5.1. The Histogram Analysis

The flatness of the histograms of pixel distributions in cipher images is a key measure of the efficiency of the encryption scheme, and confirms that crucial statistical features are suppressed by the cryptosystem. The obtained histograms for the three color components in plain shuffled and encrypted example images are illustrated in Figures 18–20.

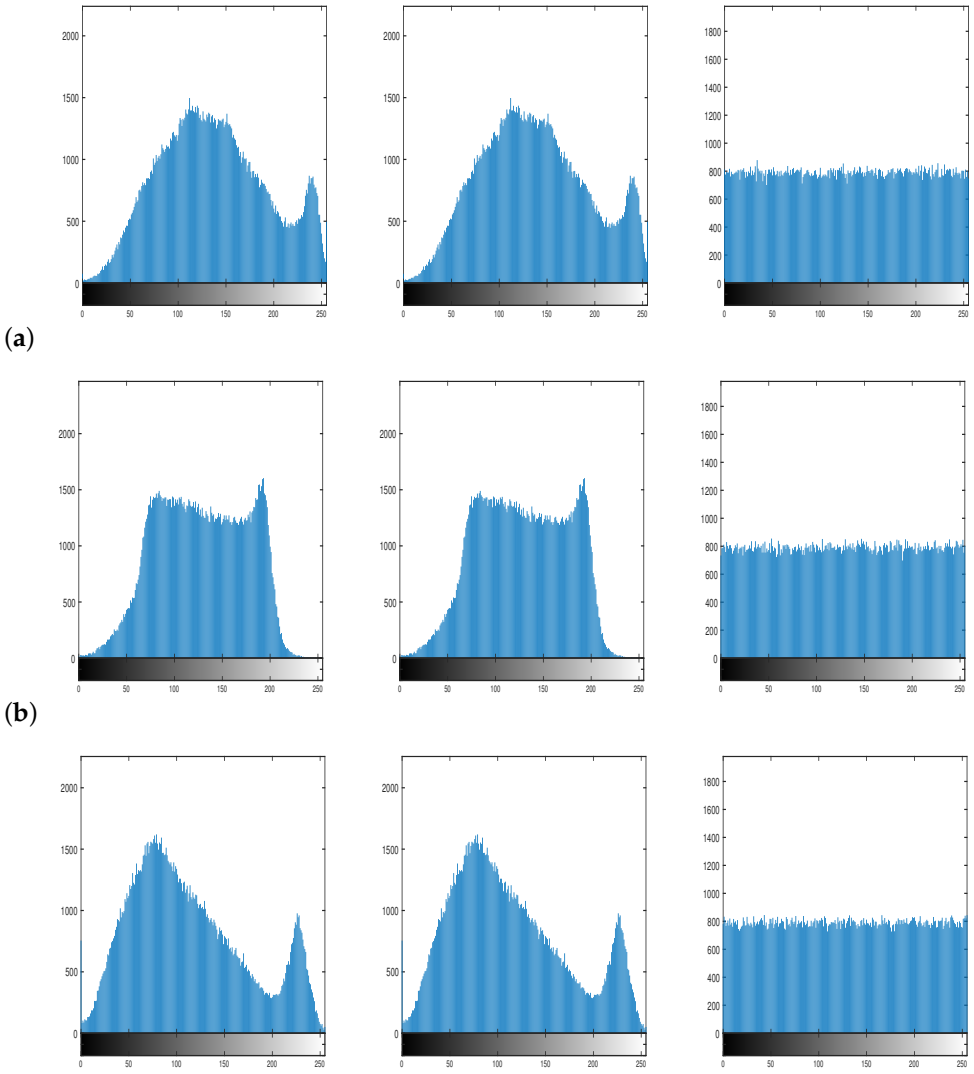


Figure 18. Histograms for (**a**) red, (**b**) green, and (**c**) blue color components of Baboon image for plain, shuffled, and cipher images, in left, middle, and right columns, respectively. The x-axis represents the pixel values of the image for the red color component (in (**a**)), green color component (in (**b**)), and blue color component (in (**c**)). The y-axis represents the number of occurrences of different pixel values in the image.

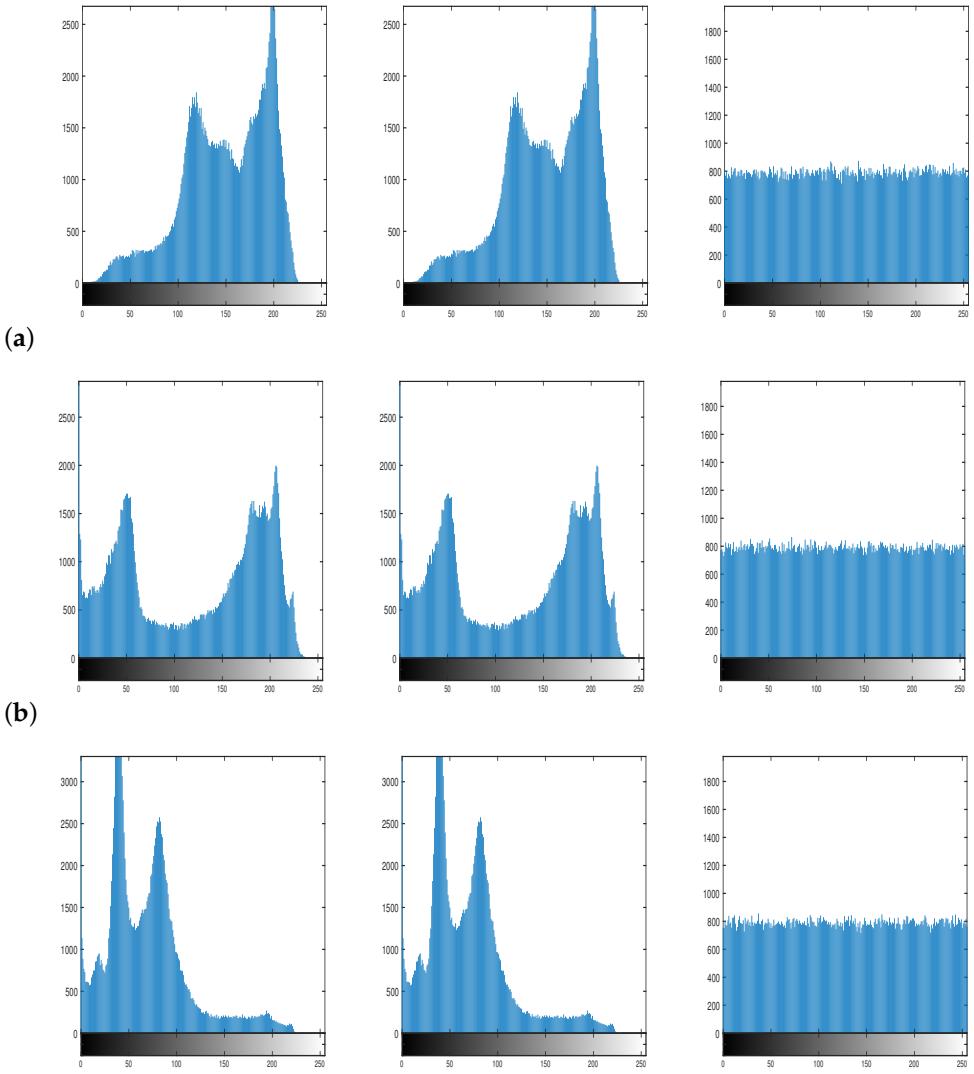


Figure 19. Similar to Figure 16 but for Pepper image.

The quantification for the flatness of the histograms can be evaluated through the variance of the histogram, which is defined by [65]

$$\Delta(H) = \frac{1}{256^2} \sum_{i=1}^{256} \sum_{j=1}^{256} \frac{1}{2} (H_i - H_j)^2, \tag{10}$$

where H denotes the the vector of histogram values, such that H_m represents the numbers of pixels with the value of m. From Table 2, it is clear that the percentage of reduction achieved in the variances of plain image and cipher image histograms is greater than 99.8% for all color components. This ensures the reliability of the proposed RC encryption technique.

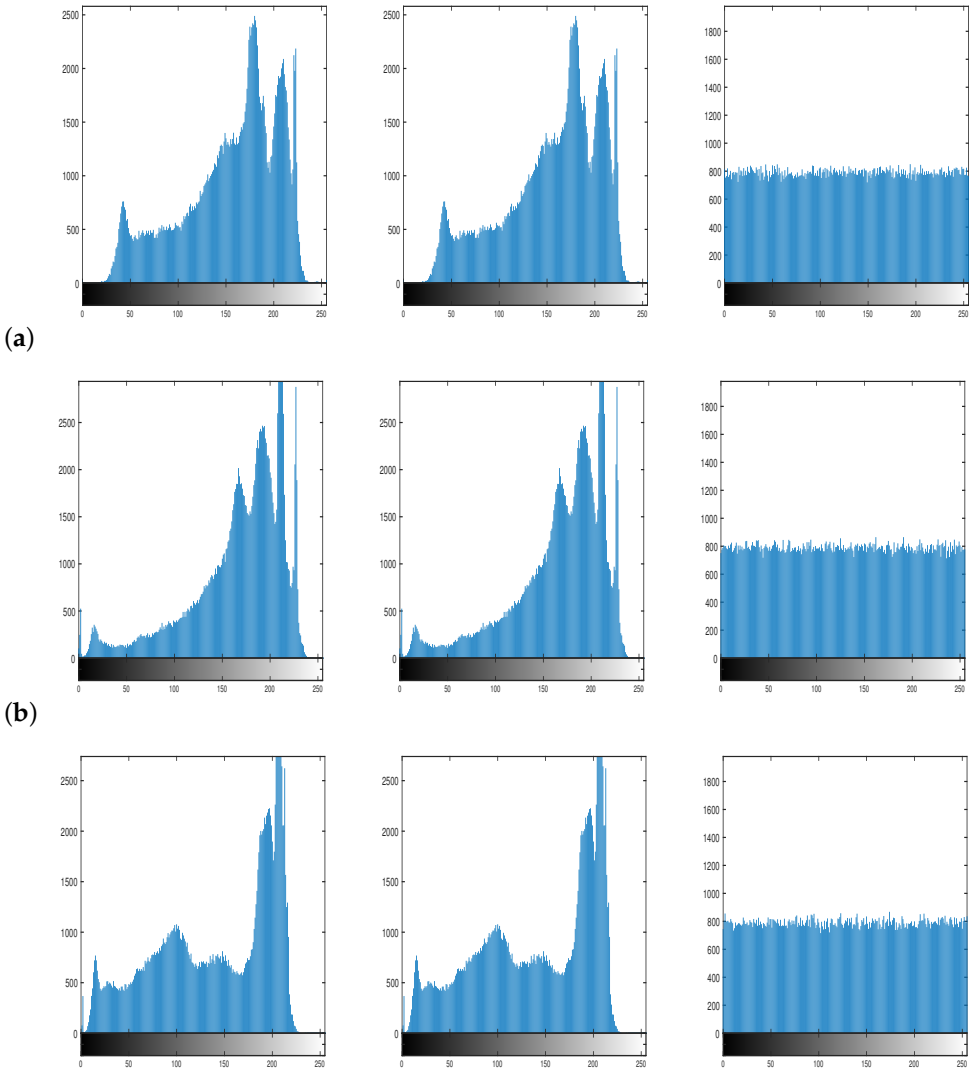


Figure 20. Similar to Figure 16 but for House image.

Table 2. The variance of histogram obtained from the three experimental images.

		Variance		
	Plain		**Encrypted**	**Reduction (%)**
Baboon	Red	176,920	349.142	99.8027
	Green	348,200	514.235	99.8524
	Blue	188,610	374.423	99.8017
Pepper	Red	520,530	775.473	99.8511
	Green	695,920	659.471	99.9082
	Blue	1,122,000	690.532	99.9385
House	Red	440,620	597.405	99.8645
	Green	756,780	728.218	99.9038
	Blue	577,050	747.452	99.8705

5.2. Analysis of Key Space

To evaluate the performance of our RC cryptosystem in opposition to possible brute force attacks, the size of its key space should be determined. It is accepted that a threshold value of 2^{100} can be considered as the minimum adequate secret key space that makes the potential brute-force attacks unfeasible [65–69]. The secret key space of the proposed RC-based encryption scheme mainly includes the $3(a+b)$ values of the training data, in addition to possible choices for perturbing vectors. This implies that for the standard binary64 IEEE 754 floating-point format, the constituted keyspace has an approximate size of $2^{159(a+b)}6!$. For $a = 50$ and $b = 50$, this implies that the secret key space is $2^{15,900}6!$, which obviously outperforms the minimum requirement of 2^{100}.

5.3. Correlation Analysis

The similarity between adjacent pixels in encrypted and plain images is investigated through correlation analysis. The reliable encryption scheme resists possible statistical attacks by greatly reducing the values of the correlation coefficients of pixels in encrypted images to be as small as possible. We define the correlation coefficient between two vectors ψ_1 and ψ_2 by

$$\kappa = \frac{cov(\psi_1, \psi_2)}{\sigma_{\psi_1}\sigma_{\psi_2}}, \tag{11}$$

where $\sigma_{\psi_1} = \sqrt{var(\psi_1)}, \sigma_{\psi_2} = \sqrt{var(\psi_2)}$ and

$$var(\psi) = \frac{1}{N}\sum_{i=1}^{N}(\psi_i - E(\psi))^2. \tag{12}$$

$$cov(\psi_1, \psi_2) = \frac{1}{N}\sum_{i=1}^{N}(\psi_{1i} - E(\psi_1))((\psi_{2i} - E(\psi_2)). \tag{13}$$

Note that the values of the adjacent pixels are referred to as ψ_1 and ψ_2. The correlation coefficients are determined in key directions in the encrypted and plain images, i.e., in the horizontal, vertical, and diagonal directions. The obtained results are depicted in Table 3, where it is shown that the present scheme successfully suppresses the values of correlation coefficients to around zero.

Table 3. The correlation coefficients of adjacent pixels in experimental images for all directions.

			Correlation Coefficients		
			Horizontal	**Vertical**	**Diagonal**
Baoon	Red	Plain	0.9193	0.864	0.8403
		Cipher	0.0004	0.0025	0.0006
	Green	Plain	0.8795	0.7997	0.7628
		Cipher	0.00052	0.0004	0.001
	Blue	Plain	0.9285	0.8827	0.8597
		Cipher	0.0007	0.0008	0.0001
Pepper	Red	Plain	0.9681	0.9703	0.9519
		Cipher	0.0001	0.00002	0.0006
	Green	Plain	0.9786	0.979	0.9616
		Cipher	0.0003	0.0019	0.0002
	Blue	Plain	0.9654	0.9643	0.9414
		Cipher	0.0013	0.0008	0.0005

Table 3. *Cont.*

Correlation Coefficients					
			Horizontal	**Vertical**	**Diagonal**
House	Red	Plain	0.9484	0.9467	0.9087
		Cipher	0.0003	0.0001	0.0004
	Green	Plain	0.9286	0.9481	0.8893
		Cipher	0.0003	0.0002	0.0003
	Blue	Plain	0.9704	0.9718	0.9472
		Cipher	0.0006	0.0003	0.0006

5.4. Information Entropy

In order to quantify the level of randomness and unpredictability in the suggested encryption technique, the information entropy analysis is employed. The information entropy for each color channel in a particular image is defined as

$$H^{r,g,b} = \sum_{i=0}^{255} p_i^{r,g,b} \log_2 \frac{1}{p_i^{r,g,b}}, \tag{14}$$

where H refers to the value of entropy in units of bits, and the probability of pixels with a value i is denoted by p_i.

The information entropy values have been obtained for the three color components in the encrypted images. The results are shown in Table 4. Knowing that the optimum value for information entropy is 8, it can be observed that the resulting information entropy from the proposed scheme is very close to this optimal value. This confirms the reliability of the RC encryption technique.

Table 4. The information entropy (bits) for different color components of encrypted images.

Plain	**Red**	**Green**	**Blue**
Baboon	7.9995	7.9994	7.9996
Pepper	7.9997	7.9998	7.9995
House	7.9995	7.9996	7.9998

5.5. Analysis of a Differential Attack

The resistance of the encryption scheme against the well-known differential attacks can be investigated by computing the NPCR (Number of Pixels Changing Rate) and the UACI (Unified Average Changing Intensity). These two quantities can precisely quantify the sensitivity of the cryptosystem to change in a single pixel in the two plain images. They can be defined as follows [65,66]

$$NPCR(\%) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \Lambda(\varsigma_1(i,j) - \varsigma_2(i,j)) \times 100, \tag{15}$$

$$UACI(\%) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|\varsigma_1(i,j) - \varsigma_2(i,j)|}{255} \times 100, \tag{16}$$

where $\Lambda(a-b) = 0$ if $a = b$, and 1, otherwise. In addition, $\varsigma_m(i,j)$ denotes the pixel value of the encrypted image m, $m = 1, 2$. Note that the associated plain images differ in a single pixel value. Table 5 illustrates the obtained results of UACI and NPCR. It is demonstrated that the NPCR values are greater than 99.6. On another side, the UACI is greater than 33.5. Thus, the proposed cryptosystem is highly sensitive to tiny changes in plain images.

Table 5. NPCR and UACI measures for cipher images of the proposed encryption scheme.

Image		$NPCR(\%)$	$UACI(\%)$
Baboon	Red	99.621	33.559
	Green	99.635	33.534
	Blue	99.631	33.541
Pepper	Red	99.647	33.512
	Green	99.617	33.539
	Blue	99.611	33.545
House	Red	99.618	33.531
	Green	99.607	33.522
	Blue	99.616	33.562

Finally, the main benefits of the proposed RC encryption scheme are summarized as follows:

(1) The proposed scheme requires much less running time, compared with encryption techniques relying on fractional order chaotic systems. Using a 16 GB RAM PC with Intel Core i7-8550U CPU @ 1.8 GHz, it takes around 1.85 s for the proposed scheme, versus 6.2 s for the conventional fractional chaotic systems-based scheme.

(2) A very large and adaptive secret key space. Indeed, the size of the key space can be easily extended to any appropriate required size.

(3) The secret keys are affected by plain images, as well as the time difference to the encryption process moment. In other words, we assume that the same plain images are to be encrypted at distinct moments. Then, different secret keys will be utilized for the proposed cryptosystem, and hence, they induce different cipher images. Then, the present scheme can efficiently resist differential attacks.

(4) As the proposed cryptosystem uses time-varying secret keys, it is immune to the powerful known-plaintext attack (KPA). More specifically, we assume that the attacker obtains some plain images and the associated cipher images. It is obvious that the opponent neither achieves his goals nor obtains any useful information regarding the values of secret keys that will be employed for other plain images.

(5) The more powerful chosen-ciphertext attack (CCA) can also be resisted by our scheme. In this attack, preselected cipher images are supplied to the receiver side to obtain the corresponding plain images. The present system is also immune to the special attack when uniform zero-pixel images are used. Note that the aforementioned attacks can cause degenerate security performance in some conventional chaos-based encryption schemes [66–69].

The comparisons with some encryption schemes in the literature are carried out in Table 6 and reveal that some schemes have a slightly larger value of NPCR or UACI than the present encryption scheme. For example, the scheme in Ref. [5] of Table 6 has NPCR = 99.679 versus NPCR = 99.621 for the suggested scheme. This implies that it is only a 0.06% increase above the NPCR value of the RC-based scheme. The other encryption schemes do not introduce a decisive increase in NPCR and UACI over our scheme.

On the other side, the secret key space of the proposed RC-based scheme achieves a 3600% (or 36 times) increase over the key space of the scheme in [5]. Indeed, the proposed cryptosystem has a distinct, very large, and adaptive key space relative to other encryption schemes. Another interesting advantage of our scheme is that it employs time-varying parameters. In other words, we assume that the same plain images are to be encrypted at different moments. The proposed scheme will use different secret keys for each time and produce different cipher images. As the proposed cryptosystem uses time-varying secret keys, it is immune to the powerful KPA attack. More specifically, when the opponent obtains some plain images and the associated cipher images, it is obvious that the attacker neither achieves his goals, nor obtains any useful information regarding the values of secret keys that will be employed for other plain images.

Table 6. Comparison with some encryption schemes in literature for the Baboon image. TVP refers to time-varying parameters and secret keys.

Work	Key Space	UACI	NPCR	Max. Corr. Coeff.	TVP
Present work	Larger than $2^{15,900}$	33.534	99.621	0.0025	YES
[3]	2^{321}	32.480	99.57	0.0210	NO
[4]	2^{372}	31.572	99.65	0.0024	NO
[5]	2^{187}	33.452	99.607	0.0082	NO
[6]	2^{624}	33.481	99.61	0.0021	NO
[7]	2^{441}	33.464	99.679	0.0013	NO

6. Conclusions

This work is an attempt to present an effective machine learning approach that is capable of predicting the time evolution of the observed nonlinear behaviors of a financial dynamical system. The well-known RNNs, such as LSTM, require large amounts of training data, and the training process is computationally expensive. The presented RC forecasting system has been verified to be a very useful tool to predict the complicated dynamics of fractional order hyperchaotic systems. Comparisons are carried out with the LSTM technique. It is found that the suggested RC-based scheme enlarges the prediction interval to at least five times greater than that of LSTM. In addition, the RC-based technique has a minimum running time (approximately 2% of the execution time of LSTM).

Moreover, a proposed RC-based encryption scheme is presented, where the RC model is employed to mimic the chaotic behavior of a prespecified nonlinear system. Compared with other encryption schemes, the proposed scheme has a very large and adaptive secret key space, which further enforces its immunity to brute force attacks. In addition, the proposed scheme requires much less running time compared with encryption techniques relying on fractional order chaotic systems. Furthrmore, the RC encryption scheme introduces time-varying secret keys, which increases its immunity against KPA, CCA, and differential attacks.

Future work can include the investigation of modified forms of RC with other types of nonlinearity, a more simple structure, a small number of nodes, and improved performance.

Author Contributions: Conceptualization, A.E.; methodology, A.E. and A.A.E.; software, W.A. and A.E.; resources, A.A.E. and A.E.; writing and review, A.E., A.A.E. and W.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research work is funded through the project number (IF2/PSAU/2022/01/21975) by the Deputyship for Research & Innovation, Ministry of Education, in Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data and material used and/or analyzed during the current study are available from the corresponding author.

Acknowledgments: The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (IF2/PSAU/2022/01/21975). The authors would like to convey their thanks to the Editor and Reviewers for their helpful comments and suggestions, which further improved this study.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

1. Chen, L.P.; Yin, H.; Yuan, L.-G.; Lopes, A.M.; Machado, J.A.T.; Wu, R.C. A novel color image encryption algorithm based on a fractional-order discrete chaotic neural network and DNA sequence operations. *Front. Inform. Technol. Elect. Eng.* **2020**, *21*, 866–879. [CrossRef]
2. Chen, Y.; Tang, C.; Ye, R. Cryptanalysis and improvement of medical image encryption using high-speed scrambling and pixel adaptive diffusion. *Signal Process.* **2020**, *167*, 107286. [CrossRef]
3. Yu, C.; Li, J.; Li, X.; Ren, X.; Gupta, B.B. Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram. *Multimed. Tools Appl.* **2018**, *77*, 585–608. [CrossRef]
4. Hamza, R.; Yan, Z.; Muhammad, K.; Bellavista, P.; Titouna, F. A privacy-preserving cryptosystem for IoT E-healthcare. *Inf. Sci.* **2020**, *527*, 493–510. [CrossRef]
5. Zhang, Y.Q.; He, Y.; Li, P.; Wang, X.Y. A new color image encryption scheme based on 2DNLCML system and genetic operations. *Opt. Lasers Eng.* **2020**, *128*, 106040. [CrossRef]
6. Gao, X.; Mou, J.; Xiong, L.; Sha, Y.; Yan, H.; Cao, Y. A fast and efficient multiple images encryption based on single-channel encryption and chaotic system. *Nonlinear Dyn.* **2022**, *108*, 613–636. [CrossRef]
7. Elsadany, A.A.; Elsonbaty, A.; Hagras, E.A. Image encryption and watermarking in ACO-OFDM-VLC system employing novel memristive hyperchaotic map. *Soft Comput.* **2023**, *12*, 1–22. [CrossRef]
8. Faster, R.C.N.N.Towards real-time object detection with region proposal networks. *Adv. Neural. Inf. Process. Syst.* **2015**, *10*, 2969239–2969250.
9. Deng, L.; Li, J.; Huang, J.T.; Yao, K.; Yu, D.; Seide, F.; Seltzer, M.; Zweig, G.; He, X.; Williams, J.; et al. Recent advances in deep learning for speech research at Microsoft. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013 ; pp. 8604–8608.
10. Chen, C.; Seff, A.; Kornhauser, A.; Xiao, J. Deepdriving: Learning affordance for direct perception in autonomous driving. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 1–18 December 2015; pp. 2722–2730. [CrossRef]
11. Kang, M.J.; Kang, J.W. Deepdriving: Intrusion detection system using deep neural network for in-vehicle network security. *PLoS ONE* **2016**, *11*, e0155781.
12. Lchen, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
13. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. India Sect. A* **1982**, *79*, 2554–2558. [CrossRef] [PubMed]
14. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
15. Maass, W.; Natschläger, T.; Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.* **2002**, *14*, 2531–2560. [CrossRef] [PubMed]
16. Jaeger, H. *The Echo State Approach to Analysing and Training Recurrent Neural Networks-with an Erratum Note*; German National Research Center for Information Technology GMD Technical Report; German National Research Center: Bonn, Germany, 2001; Volume 148, p. 13.
17. Lukosevcius, M.; Jaeger, H.; Schrauwen, B. Reservoir computing trends. *KI-KüNstliche Intell.* **2012**, *26*, 365–371. [CrossRef]
18. Lukosevcius, M.; Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **2009**, *3*, 127–149. [CrossRef]
19. Tanaka, G.; Yamane, T.; Héroux, J.B.; Nakane, R.; Kanazawa, N.; Takeda, S.; Hirose, A. Recent advances in physical reservoir computing: A review. *Neural Netw.* **2019**, *115*, 100–123. [CrossRef]
20. Schrauwen, B.; Verstraeten, D.; Van Campenhout, J. An overview of reservoir computing: Theory, applications and implementations. In Proceedings of the 15th European Symposium on Artificial Neural Networks, Bruges, Belgium, 25–27 April 2007; pp. 471–482.
21. Gallicchio, G.; Micheli, A.; Pedrelli, L. Deep reservoir computing: A critical experimental analysis. *Neurocomputing* **2017**, *268*, 87–99. [CrossRef]
22. Paugam-Moisy, H.; Martinez, R.; Bengio, S. Delay learning and polychronization for reservoir computing. *Neurocomputing* **2008**, *71*, 1143–1158. [CrossRef]
23. Butcher, J.B.; Verstraeten, D.; Schrauwen, B.; Day, C.R.; Haycock, P.W. Reservoir computing and extreme learning machines for non-linear time-series data analysis. *Neural Netw.* **2013**, *38*, 76–89. [CrossRef]
24. Jaeger, H.; Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* **2004**, *304*, 78–80. [CrossRef]
25. Liu, J.; Zhang, J.; Wang, Y. Secure communication via chaotic synchronization based on reservoir computing. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–15 . [CrossRef] [PubMed]
26. Ferreira, A.A.; Ludermir, T.B.; de Aquino, R.R.; Lira, M.M.; Neto, O.N. Investigating the use of reservoir computing for forecasting the hourly wind speed in short-term. In Proceedings of the IEEE International Joint Conference on Neural Network, Hong Kong, China, 1–8 June 2008; pp. 1649–1656.
27. Escalona-Moran, M.A.; Soriano, M.C.; Fischer, I.; Mirasso, C.R. Electrocardiogram classification using reservoir computing with logistic regression. *IEEE J. Biomed. Health Inform.* **2014**, *19*, 892–898. [CrossRef] [PubMed]

28. Jalalv, A.; Demuynck, K.; De Neve, W.; Martens, J.P. On the application of reservoir computing networks for noisy image recognition. *Neurocomputing* **2018**, *277*, 237–248. [CrossRef]
29. Verstraeten, D.; Schrauwen, B.; Stroob, T.D. Reservoir-based techniques for speech recognition. In Proceedings of the IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada, 16–21 July 2006; pp. 1050–1053. [CrossRef]
30. Martinenghi, R.; Rybalko, S.; Jacquot, M.; Chembo, Y.K.; Larger, L. Photonic nonlinear transient computing with multiple-delay wavelength dynamics. *Phys. Rev. Lett.* **2012**, *108*, 244101. [CrossRef] [PubMed]
31. Vandoorne, K.; Mechet, P.; Van Vaerenbergh, T.; Fiers, M.; Morthier, G.; Verstraeten, D.; Schrauwen, B.; Dambre, J.; Bienstman, P. Experimental demonstration of reservoir computing on a silicon photonics chip. *Nat. Comm.* **2012**, *5*, 3541. [CrossRef] [PubMed]
32. Antonik, P.; Duport, F.; Hermans, M.; Smerieri, A.; Haelterman, M.; Massar, S. Online training of an opto-electronic reservoir computer applied to real-time channel equalization. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2686–2698. [CrossRef]
33. Du, C.; Cai, F.; Zidan, M.A.; Ma, W.; Lee, S.H.; Lu, W.D. Reservoir computing using dynamic memristors for temporal information processing. *Nat. Comm.* **2017**, *8*, 2204. [CrossRef]
34. Moon, J.; Ma, W.; Shin, J.H.; Cai, F.; Du, C.; Lee, S.H.; Lu, W.D. Temporal data classification and forecasting using a memristor-based reservoir computing system. *Nat. Electron.* **2019**, *2*, 480–487. [CrossRef]
35. Midya, R.; Wang, Z.; Asapu, S.; Zhang, X.; Rao, M.; Song, W.; Yang, J.J. Reservoir computing using diffusive memristors. *Adv. Intell. Syst.* **2019**, *1*, 1900084. [CrossRef]
36. Yao, P.; Wu, H.; Gao, B.; Eryilmaz, S.B.; Huang, X.; Zhang, W.; Qian, H. Face classification using electronic synapses. *Nat. Comm.* **2017**, *8*, 15199. [CrossRef]
37. Hu, M.; Graves, C.E.; Li, C.; Li, Y.; Ge, N.; Montgomery, E.; Strachan, J.P. Memristor-based analog computation and neural network classification with a dot product engine. *Adv. Mater.* **2018**, *30*, 1705914. [CrossRef]
38. Yang, J.J.; Strukov, D.B.; Stewart, D.R. Memristive devices for computing. *Nat. Nanotechnol.* **2013**, *8*, 13–24. [CrossRef] [PubMed]
39. Cai, F.; Correll, J.M.; Lee, S.H.; Lim, Y.; Bothra, V.; Zhang, Z.; Lu, W.D. A fully integrated reprogrammable memristor CMOS system for efficient multiply accumulate operations. *Nat. Electro.* **2019**, *2*, 290–299. [CrossRef]
40. Wang, W.J.; Tang, Y.; Xiong, J.; Zhang, Y.C. Stock market index prediction based on reservoir computing models. *Expert Syst. Appl.* **2021**, *178*, 115022. [CrossRef]
41. Budhiraja, R.; Kumar, M.; Das, M.K.; Bafila, A.S.; Singh, S. A reservoir computing approach for forecasting and regenerating both dynamical and time-delay controlled financial system behavior. *PLoS ONE* **2021**, *16*, e0246737. [CrossRef] [PubMed]
42. Wyffels, F.; Schrauwen, B. A comparative study of reservoir computing strategies for monthly time series prediction. *Neurocomputing* **2010**, *73*, 1958–1964. [CrossRef]
43. Gonon, L.; Grigoryea, L.; Ortega, J.P. Risk bounds for reservoir computing. Journal of Machine Learning Research. *J. Mach. Learn. Res.* **2020**, *21*, 9684–9744
44. Der Wang, T.; Wu, X.; Fyfe, C. Comparative, study, of, visualisation, methods, for, temporal, data. In Proceedings of the IEEE Congress on Evolutionary Computation, Brisbane, Australia, 10–15 June 2012; pp. 1–5. [CrossRef]
45. Szuminski, W. Integrability analysis of chaotic and hyperchaotic finance systems. *Nonlinear Dyn.* **2018**, *94*, 443–459. [CrossRef]
46. Yu, H.; Cai, G.; Li, Y. Dynamic analysis and control of a new hyperchaotic finance system. *Nonlinear Dyn.* **2012**, *67*, 2171–2182. [CrossRef]
47. Vargas, J.A.; Grzeidak, E.; Hemerly, E.M. Robust adaptive synchronization of a hyperchaotic finance system. *Nonlinear Dyn.* **2015**, *80*, 239–248. [CrossRef]
48. Cao, L. A four-dimensional hyperchaotic finance system and its control problems. *J. Control Sci. Eng.* **2018**, *2018*, 4976380. [CrossRef]
49. Ding, J.; Yang, W.; Yao, H. A new modified hyperchaotic finance system and its control. *Int. J. Nonlinear Sci.* **2009**, *8*, 59–66.
50. Chen, C.; Fan, T.; Wang, B. Inverse optimal control of hyperchaotic finance system. *WJMS* **2014**, *10*, 83–91.
51. Tong, X.J.; Zhang, M.; Wang, Z.; Liu, Y.; Ma, J. An image encryption scheme based on a new hyperchaotic finance system. *Optik* **2015**, *126*, 2445–2452. [CrossRef]
52. Kocamaz, U.E.; Göksu, A.; Uyaroglu, Y.; Taskin, H. Controlling hyperchaotic finance system with combining passive and feedback controllers. *Inf. Technol. Control.* **2018**, *47*, 45–55. [CrossRef]
53. Jahanshai, H.; Yousefpour, A.; Wei, Z.; Alcaraz, R.; Bekiros, S. A financial hyperchaotic system with coexisting attractors: Dynamic investigation, entropy analysis, control and synchronization. *Chaos Solitons Fractals* **2019**, *126*, 66–77. [CrossRef]
54. Hajipour, A.; Hajipour, M.; Baleanu, D. On the adaptive sliding mode controller for a hyperchaotic fractional-order financial system. *Phys. A: Stat. Mech. Appl.* **2018**, *497*, 139–153. [CrossRef]
55. Chen, H.; Yu, L.; Wang, Y.; Guo, M. Synchronization of a hyperchaotic finance system. *Complexity* **2021**, 2021, 6618435. [CrossRef]
56. Bekiros, S.; Jahanshahi, H.; Bezzina, F.; Aly, A.A. A novel fuzzy mixed H2/H8 optimal controller for hyperchaotic financial systems. *Chaos Solitons Fractals* **2021**, *146*, 110878. [CrossRef]
57. Kumar, S.; Prasad, R.P.; Pal, K.; Pal, M.P.; Singh, A. Synchronization of Fractional-Order Hyperchaotic Finance Systems Using Sliding Mode Control Techniques, IGI global. In *Advanced Applications of Fractional Differential Operators to Science and Technology*; IGI Global: Hershey, PA, USA, 2020; pp. 133–152.
58. Lazopoulos, K.A.; Lazopoulos, A.K. Fractional vector calculus and fluid mechanics. *J. Mech. Behav. Biomed. Mater.* **2017**, *26*, 43–54. [CrossRef]

59. Diouf, M.; Sene, N. Analysis of the financial chaotic model with the fractional derivative operator. *Complexity* **2020**, 2020, 9845031. [CrossRef]
60. Koeller, R. Applications of fractional calculus to the theory of viscoelasticity. *J. Appl. Mecha.* **1984**, *51*, 299–307. [CrossRef]
61. Senthilkumar, V. Fractional Derivative Analysis of Wave Propagation Studies Using Eringen's Nonlocal Model with Elastic Medium Support. *J. Vib. Eng. Technol.* **2022**, 1–9 . [CrossRef]
62. Sierociuk, D.; Skovranek, T.; Macias, M.; Podlubny, I.; Petras, I.; Dzielinski, A.; Ziubinski, P. Diffusion process modeling by using fractional-order models. *Appl. Math. Comput.* **2015**, *257*, 2–11. [CrossRef]
63. Chimmula, V.; Zhang, L. Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos Solitons Fractals* **2020**, *135*, 109864. [CrossRef] [PubMed]
64. Elsheikh, A.H.; Saba, A.I.; Elaziz, M.A.; Lu, S.; Shanmugan, S.; Muthuramalingam, T.; Kumar, R.; Mosleh, A.O.; Essa, F.A.; Shehabeldeen, T.A. Deep learning-based forecasting model for COVID-19 outbreak in Saudi Arabia. *Process. Saf. Environ. Prot.* **2021**, *149*, 223–233. [CrossRef] [PubMed]
65. Al-Khedhairi, A.; Elsonbaty, A.; Elsadany, A.A.; Hagras, E.A. Hybrid cryptosystem based on pseudo chaos of novel fractional order map and elliptic curves. *IEEE Access* **2020**, *8*, 57733–57748. [CrossRef]
66. Zhang, Y.Q.; Wang, X.Y. A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice. *Inform. Sci.* **2014**, *273*, 329–351. [CrossRef]
67. Lin, Z.; Yu, S.; Feng, X.; Lü, J. Cryptanalysis of a chaotic stream cipher and its improved scheme. *IJBC* **2018**, *28*, 1850086. [CrossRef]
68. Xiao, D.; Liao, X.; Wei, P. Analysis and improvement of a chaos-based image encryption algorithm. *Chaos Solitons Fractals* **2009**, *40*, 2191–2199. [CrossRef]
69. Li, C.; Lin, D.; Feng, B.; Lü, J.; Hao, F. Cryptanalysis of a chaotic image encryption algorithm based on information entropy. *IEEE Access* **2018**, *6*, 75834–75842. [CrossRef]