ISSN: 2687 - 4539

CHAOS THEORY AND APPLICATIONS

IN APPLIED SCIENCES AND ENGINEERING

This issue is dedicated to the memory of Prof. Tenreiro Machado.

VOLUME 4, ISSUE 2, JULY 2022

AN INTERDISCIPLINARY JOURNAL OF NONLINEAR SCIENCE

ISSN: 2687 - 4539

CHAOS
THEORY AND APPLICATIONS
IN APPLIED SCIENCES AND ENGINEERING

Chaos Theory and Applications (CHTA)
Volume: 4 – Issue No: 2 (July 2022)
https://dergipark.org.tr/en/pub/chaos/issue/64884

ISSN: 2687 - 4539

CHAOS

THEORY AND APPLICATIONS

IN APPLIED SCIENCES AND ENGINEERING

Chaos Theory and Applications (CHTA)
Volume: 4 – Issue No: 2 (July 2022)
https://dergipark.org.tr/en/pub/chaos/issue/64884

Contents

CHAOS Theory and Applications in Applied Sciences and Engineering

e-ISSN: 2687-4539
EDITORIAL
Vol.4 / No.2 / 2022 / pp.59-63

The Evolution of Fractional Calculus

J. A. Tenreiro Machado *,1
*Institute of Engineering of Porto, Polytechnic of Porto, Portugal.

ABSTRACT Fractional Calculus started in 1695 with Leibniz discussing the meaning of $D^n y$ for $n = 1/2$. Many mathematicians developed the theoretical concepts, but the area remained somewhat unknown from applied sciences. During the eighties FC emerged associated with phenomena such as fractal and chaos and, consequently, in nonlinear dynamical. In the last years, Fractional Calculus became a popular tool for the modeling of complex dynamical systems with nonlocality and long memory effects.

KEYWORDS
Fractional calculus
Non-locality
Long range memory

INTRODUCTION

The generalization of the concept of derivative $D^\alpha f(x)$ to non-integer values of α goes back to the beginning of the theory of differential calculus in the follow-up of the brilliant ideas of Gottfried Leibniz (Machado and Kiryakova 2019). The development of this area of knowledge is due to the contributions of important scientists such as Euler, Liouville and Riemann (Machado *et al.* 2010; Valério *et al.* 2014) as represented in Fig. 1. In the fields of physics and engineering, Fractional Calculus (FC) is presently associated with the modelling of complex phenomena with nonlocality and long memory effects (Tarasov 2019a,b; Băleanu and Lopes 2019a,b). This paper introduces the fundamentals of this tool, its application in the control of dynamical systems, and present day state of development.

MATHEMATICAL FUNDAMENTALS OF FRACTIONAL CALCULUS

The most used definitions of a fractional derivative of order α are the Riemann-Liouville (RL, $t > a$, $Re(\alpha) \in]n-1, n[$), Grünwald-Letnikov (GL, $t > a$, $\alpha > 0$) and Caputo (C, $t > a$, $n-1 < \alpha < n$) formulations (Kochubei and Luchko 2019a,b; Karniadakis 2019):

$$ {}_a^{RL}D_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau, \tag{1a}$$

$$ {}_a^{GL}D_t^\alpha f(t) = \lim_{h \to 0} \frac{1}{h^\alpha} \sum_{k=0}^{\left[\frac{t-a}{h}\right]} (-1)^k \begin{pmatrix} \alpha \\ k \end{pmatrix} f(t-kh), \tag{1b}$$

$$ {}_a^{C}D_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \int_a^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau, \tag{1c}$$

where $\Gamma(\cdot)$ is Euler's gamma function, $[x]$ means the integer part of x, and h is the step time increment.

These operators capture the history of all past events, in opposition to integer derivatives that are 'local' operators. This means that fractional order systems have a memory of the dynamical evolution. This behaviour has been recognized in several natural and man made phenomena and their modelling becomes much simpler using the tools of FC, while the counterpart of building integer order models leads often to complicated expressions Machado and Lopes (2020b,a). The geometrical interpretation of fractional derivatives has been the subject of debate and several perspectives have been proposed (Machado 2003, 2021).

Manuscript received: 9 September 2021,
Accepted: 11 September 2021.

[1] jtm@isep.ipp.pt (**Corresponding author**)

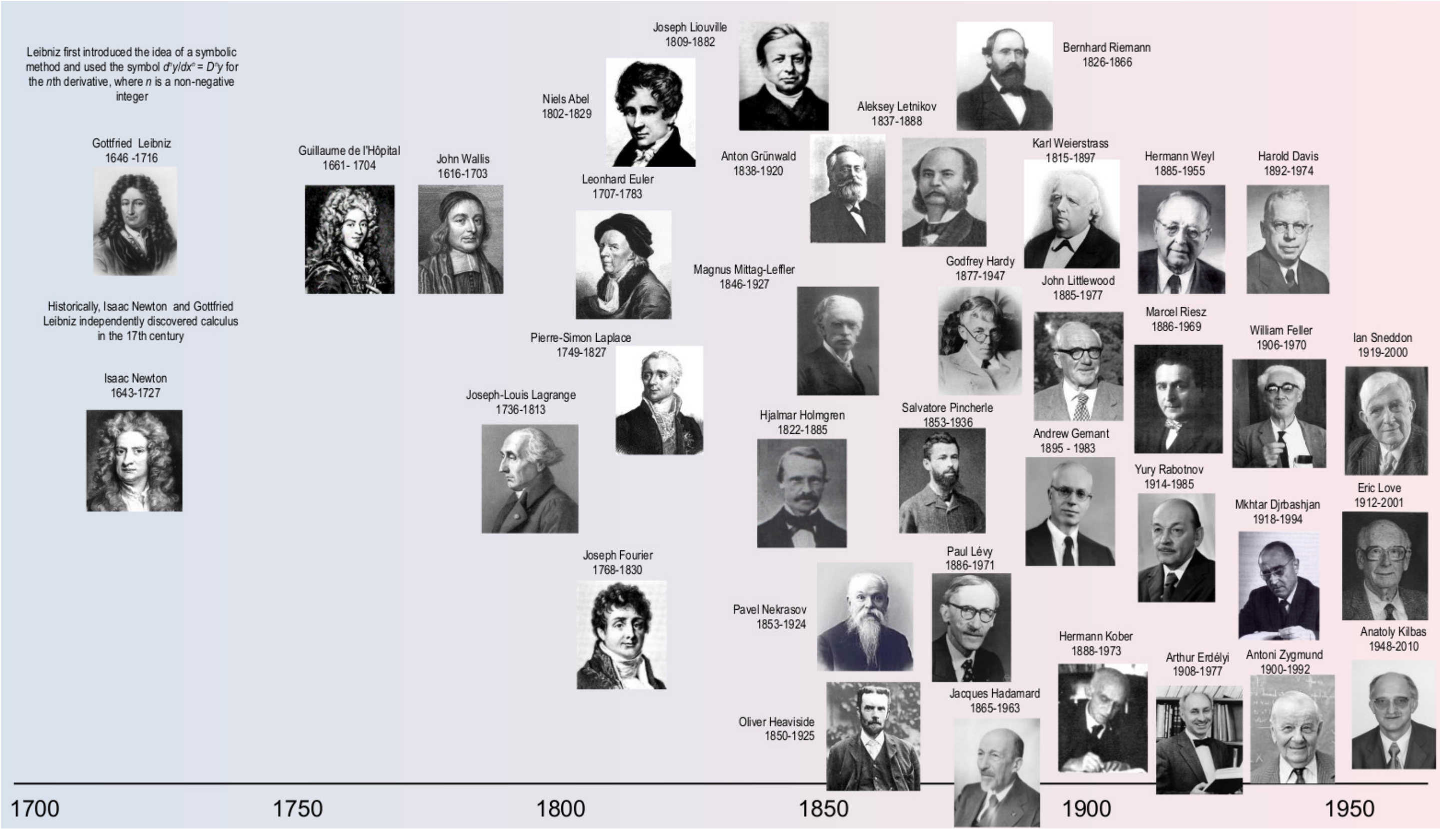


Figure 1 The FC timeline

Using the Laplace transform we have the expressions:

$$\mathcal{L}\left\{{}_0^{RL}D_t^{\alpha} f(t)\right\} = s^{\alpha}\mathcal{L}\left\{f(t)\right\} - \sum_{k=0}^{n-1} s^k \, {}_0^{RL}D_t^{\alpha-k-1} f\left(0^+\right), \tag{2a}$$

$$\mathcal{L}\left\{{}_0^{C}D_t^{\alpha} f(t)\right\} = s^{\alpha}\mathcal{L}\left\{f(t)\right\} - \sum_{k=0}^{n-1} s^{\alpha-k-1} \, f^{(k)}(0), \tag{2b}$$

where s and $\mathcal{L}$ denote the Laplace variable and operator, respectively.

The Mittag-Leffler function (MLF), $E_{\alpha}(t)$, is defined as:

$$E_{\alpha}(t) = \sum_{k=0}^{\infty} \frac{t^k}{\Gamma(\alpha k + 1)}, \; \alpha \in \mathbb{C}, \; \mathrm{Re}(\alpha) > 0. \tag{3}$$

The MLF represents a bridge between the exponential and the power law functions. In particular, when $\alpha = 1$ the MLF simplifies and we have $E_1(t) = e^t$, while, for large values of t, the asymptotic behaviour yields $E_{\alpha}(-t) \approx \frac{1}{\Gamma(1-\alpha)}\frac{1}{t}$, $\alpha \neq 1, 0 < \alpha < 2$.

Since the Laplace transform leads to:

$$\mathcal{L}\left\{E_{\alpha}\left(\pm a t^{\alpha}\right)\right\} = \frac{s^{\alpha-1}}{s^{\alpha} \mp a} \tag{4}$$

we observe a generalization of the Laplace transform pairs from the exponential towards the ML, namely from integer up to fractional powers of s. The more general MLF, often called two-parameter MLF, is given by:

$$E_{\alpha,\beta}(t) = \sum_{k=0}^{\infty} \frac{t^k}{\Gamma(\alpha k + \beta)}, \; \alpha, \beta \in \mathbb{C}, \; \mathrm{Re}(\alpha), \mathrm{Re}(\beta) > 0. \tag{5}$$

The function defined by (3) gives a generalization of (5), since $E_{\alpha}(t) = E_{\alpha,1}(t)$.

FRACTIONAL CONTROL

Let us consider an elemental feedback control system of fractional order α, with unit feedback and transfer function $G(s) = \frac{K}{s^{\alpha}}$, $1 < \alpha < 2$, in the direct loop (Machado 1997, 2001). The open-loop Bode diagrams of amplitude and phase have a slope of -20 dB/dec and a constant phase of $-\alpha\frac{\pi}{2}$ rad, respectively. Therefore, the closed-loop system has a constant phase margin of $\pi\left(1 - \frac{\alpha}{2}\right)$ rad, that is independent of the system gain K.

Assume that $K = 1$, so that $G(s) = \frac{1}{s^\alpha}$, and that the closed-loop system is excited by an unit step input $R(s) = \frac{1}{s}$. The output response will be $C(s) = \frac{1}{s(s^\alpha+1)}$, or, in the time domain, $c(t) = 1 - E_\alpha(-t^\alpha)$. Figure 2 depicts the responses for $\alpha = \{0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2\}$. We observe that the fractional values 'interpolate' the cases of integer orders $\alpha = \{0, 1, 2\}$. We note a fast initial transient followed by a slow convergence for the steady-state value, which is typical of many fractional order systems.

A popular application of FC is in the area of control (Petráš 2019) and corresponds to the generalization of the Proportional, Integral and Derivative (PID) algorithm, namely to the fractional PID. The $PI^\lambda D^\mu$ control algorithm has a transfer function given by:

$$G_c(s) = K_P + K_I s^{-\lambda} + K_D s^\mu, \tag{6}$$

where K_P, K_I and K_D are the proportional, integral and differential gains, and λ and μ are the fractional orders of the integral and derivative actions, respectively. The cases $(\lambda, \mu) = \{(0,0), (1,0), (0,1), (1,1)\}$, correspond to the P, PI, PD and PID, respectively.

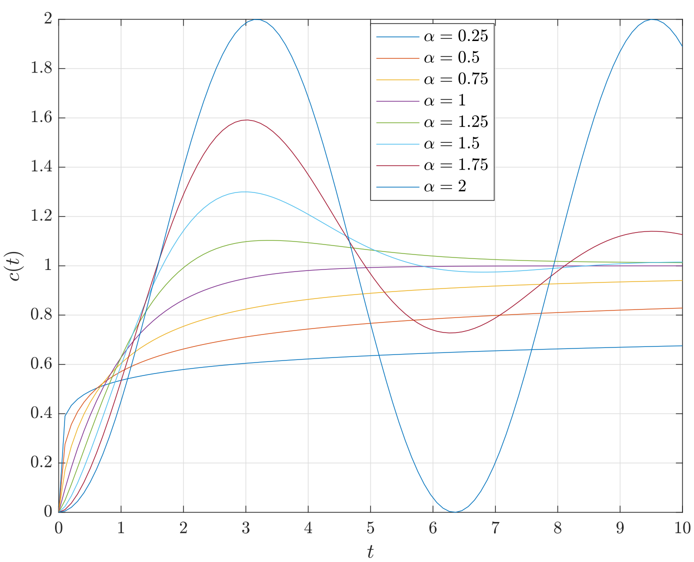


Figure 2 Time response $c(t) = 1 - E_\alpha(-t^\alpha)$ of the fractional closed-loop system for a unit step reference input and $\alpha = \{0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2\}$

PROGRESS OF THE FRACTIONAL CALCULUS

We can estimate of the present day state of FC using publicly available information, just to remind that until 1974 there were only 1 book devoted to FC as a topic, while by 2018 the number of FC books were estimated to be more than 240 Machado and Kiryakova (2017). For that purpose we selected the program VOSviewer van Eck and Waltman (2009, 2017) as the tool for processing bibliographic information.

Let us consider (i) data is available at Scopus database, (ii) papers published during year 2020, and (iii) 8 search keywords, namely {Fractional calculus, Fractional derivative, Fractional integration, Fractional dynamics, Mittag-Leffler, Derivative of non-integer order, Integral of non-integer order, Derivative of complex order, Integral of complex order} that yields 6,589 records. The VOSViewer allows several perspectives of bibliographic analysis, but let us start by considering a network plot for the options 'Co-occurrence', 'All keywords', 'Full counting', 'Minimum number of occurrence of a keyword=4'. This search gives 2,764 keywords, as shown in Fig. 3. On the other hand Fig. 4 depicts the network plot for the options 'Co-authotship', 'Countries', 'Full counting', 'Minimum number of occurrence of a country=4', , 'Minimum number of citations of a country=2' that gives 77 cases. The two network plots show that FC is presently applied in all fields of science, going from the areas of mathematics, physics, engineering and economy, up to medicine, biology and genetics, and the topic is presently very popular in all countries of the globe.

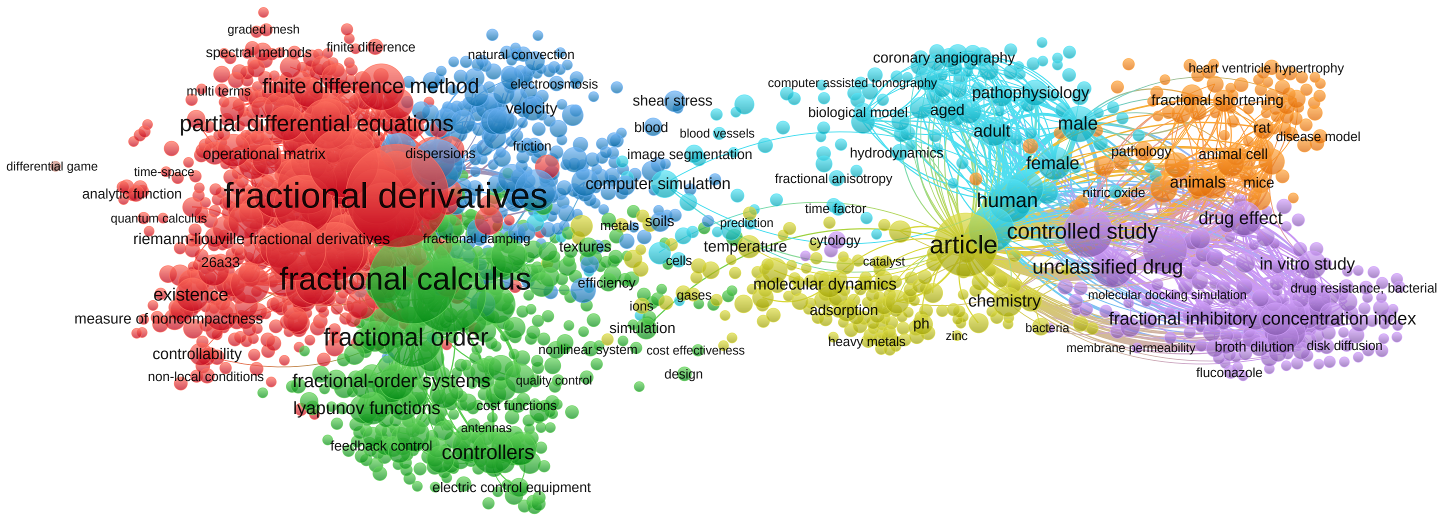


Figure 3 VOSviewer network plot with options ‘Co-occurrence’, ‘All keywords’, ‘Full counting’, ‘Minimum number of occurrence of a keyword=4’

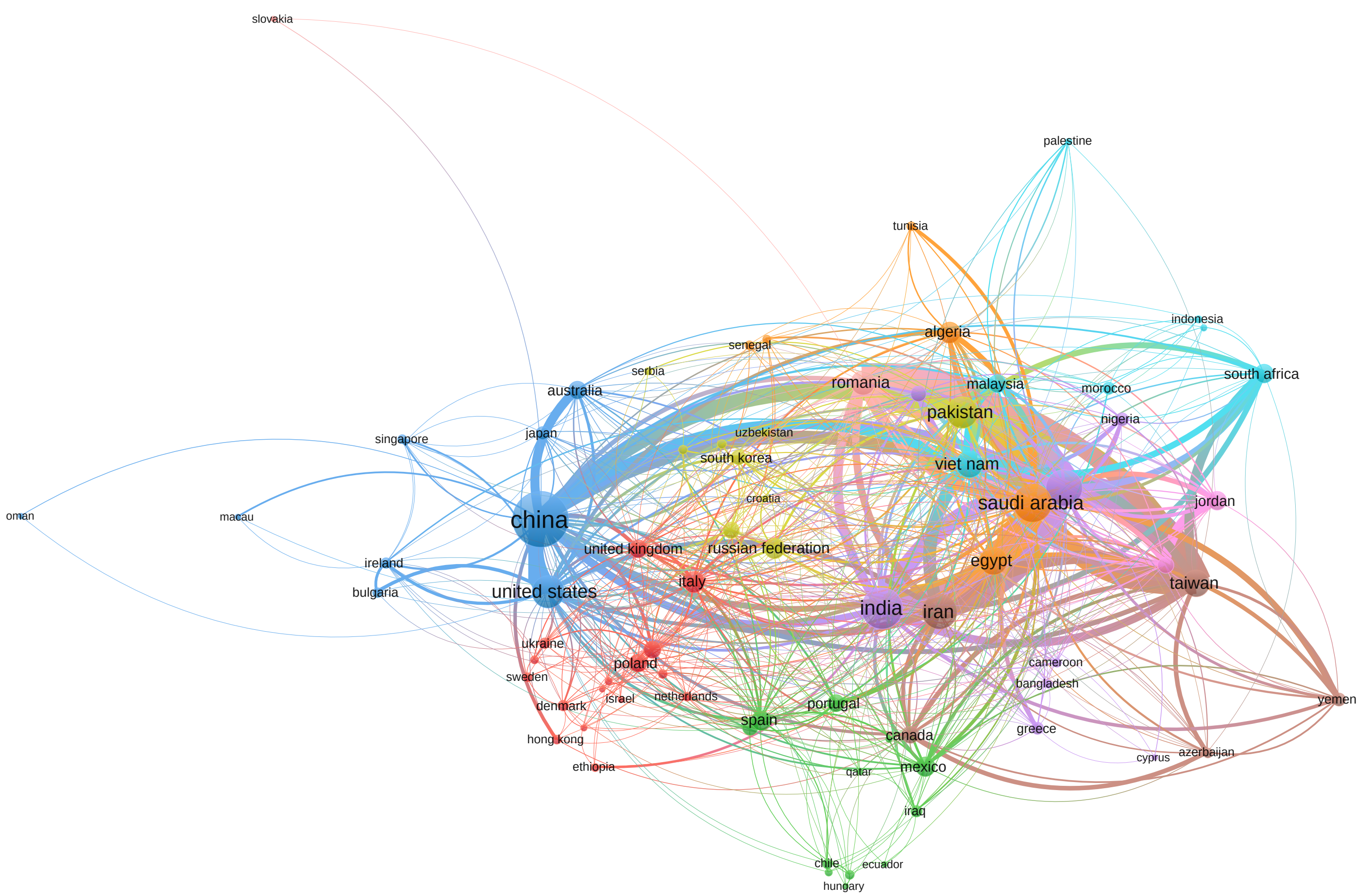


Figure 4 VOSviewer network plot with options ‘Co-occurrence’, ‘All keywords’, ‘Full counting’, ‘Minimum number of occurrence of a country=4’, , ‘Minimum number of citations of a country=2’

CONCLUSIONS

This work introduced and discussed several aspects of the FC. The history, fundamentals and the use of FC in control were described. The present day areas of application of FC and its evolution were also analyzed using a computer package for processing bibliographic information.

Conflicts of interest

The author declares that there is no conflict of interest regarding the publication of this paper.

LITERATURE CITED

Băleanu, D. and A. M. Lopes, editors, 2019a *Handbook of Fractional Calculus with Applications: Applications in Engineering, Life and Social Sciences, Part A*, volume 7 of *De Gruyter Reference*. De Gruyter, Berlin.

Băleanu, D. and A. M. Lopes, editors, 2019b *Handbook of Fractional Calculus with Applications: Applications in Engineering, Life and Social Sciences, Part B*, volume 8 of *De Gruyter Reference*. De Gruyter, Berlin.

Karniadakis, G. E., editor, 2019 *Handbook of Fractional Calculus with Applications: Numerical Methods*, volume 3 of *De Gruyter Reference*. De Gruyter, Berlin.

Kochubei, A. and Y. Luchko, editors, 2019a *Handbook of Fractional Calculus with Applications: Basic Theory*, volume 1 of *De Gruyter Reference*. De Gruyter, Berlin.

Kochubei, A. and Y. Luchko, editors, 2019b *Handbook of Fractional Calculus with Applications: Fractional Differential Equations*, volume 2 of *De Gruyter Reference*. De Gruyter, Berlin.

Machado, J. and V. Kiryakova, 2017 The chronicles of fractional calculus. Fractional Calculus and Applied Analysis **20**: 307–336.

Machado, J. A. T., 2003 A probabilistic interpretation of the fractional-order differentiation. Fractional Calculus & Applied Analysis **6**: 73–80.

Machado, J. A. T., 2021 The bouncing ball and the Grünwald-Letnikov definition of fractional derivative. Fractional Calculus and Applied Analysis **24**: 1003–1014.

Machado, J. A. T. and V. Kiryakova, 2019 Recent history of the fractional calculus: data and statistics. In *Handbook of Fractional Calculus with Applications: Basic Theory*, edited by A. Kochubei and Y. Luchko, pp. 1–22, De Gruyter.

Machado, J. A. T., V. Kiryakova, and F. Mainardi, 2010 A poster about the recent history of fractional calculus. Fractional Calculus and Applied Analysis **13**: 329–334.

Machado, J. T., 1997 Analysis and design of fractional-order digital control systems. Systems Analysis, Modelling, Simulation **27**: 107–122.

Machado, J. T., 2001 Discrete-time fractional-order controllers. Fractional Calculus & Applied Analysis **4**: 47–66.

Machado, J. T. and A. M. Lopes, 2020a Multidimensional scaling and visualization of patterns in prime numbers. Communications in Nonlinear Science and Numerical Simulation **83**: 105128.

Machado, J. T. and A. M. Lopes, 2020b Multidimensional scaling locus of memristor and fractional order elements. Journal of Advanced Research **25**: 147–157.

Petráš, I., editor, 2019 *Handbook of Fractional Calculus with Applications: Applications in Control*, volume 6 of *De Gruyter Reference*. De Gruyter, Berlin.

Tarasov, V. E., editor, 2019a *Handbook of Fractional Calculus with Applications: Applications in Physics, Part A*, volume 4 of *De Gruyter Reference*. De Gruyter, Berlin.

Tarasov, V. E., editor, 2019b *Handbook of Fractional Calculus with Applications: Applications in Physics, Part B*, volume 5 of *De Gruyter Reference*. De Gruyter, Berlin.

Valério, D., J. Machado, and V. Kiryakova, 2014 Some pioneers of the applications of fractional calculus. Fractional Calculus and Applied Analysis **17**: 552–578.

van Eck, N. J. and L. Waltman, 2009 Software survey: VOSviewer, a computer program for bibliometric mapping. Scientometrics **84**: 523–538.

van Eck, N. J. and L. Waltman, 2017 Citation-based clustering of publications using CitNetExplorer and VOSviewer. Scientometrics **111**: 1053–1070.

How to cite this article: Machado, J. A. T. *The Evolution of Fractional Calculus*. Chaos Theory and Applications, 4(2), 59-63, 2022.

CHAOS Theory and Applications in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.4 / No.2 / 2022 / pp.64-70
https://doi.org/10.51537/chaos.1112243

Experimental Validation of a Chaotic Jerk Circuit Based True Random Number Generator

R. Chase Harrison[*,1], **Benjamin K. Rhea**[*,2], **Ariel R. Oldag**[*,3], **Robert N. Dean**[*,4] **and Edmon Perkins**[†,5]

[*]Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849, United States, [†]LAB2701, Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC 27695, United States.

ABSTRACT A method for true random number generation by directly sampling a high frequency chaotic jerk circuit is explored. A method for determination of the maximum Lyapunov exponent, and thus the maximum bit rate for true random number generation, of the jerk system of interest is shown. The system is tested over a wide range of sampling parameters in order to simulate possible hardware configurations. The system is then implemented in high speed electronics on a small printed circuit board to verify its performance over the chosen parameters. The resulting circuit is well suited for random number generation due to its high dynamic complexity, long term aperiodicity, and extreme sensitivity to initial conditions. This system passes the Dieharder RNG test suite at 3.125 Mbps.

KEYWORDS
Random number generation
Diehard
Chaotic circuits
Chaos theory
Nonlinear dynamics
Jerk oscillator

INTRODUCTION

Chaos and randomness have gone hand-in-hand since the inception of the idea that both natural and man-made systems could produce wildly different behaviors given seemingly identical conditions. There has always been a struggle to determine the outcome of future processes given only present information, but the feasibility of these endeavors has only recently been quantified in terms of entropy and randomness. Efforts to achieve this understanding have shed light on the inherent nonlinear dynamics. As such, these properties can be exploited to achieve a randomness that can be understood and measured, yet retain the useful unpredictable nature of these dynamics.

Security and communications systems today, including financial security, RFID, and cryptography, rely on this idea of randomness Sundaresan *et al.* (2015); Volos (2013). Specifically, these technologies depend heavily on random bits being readily available to process into various encryption schemes. It is important that the random numbers in these systems exhibit various statistical properties that are indicative of a theoretically perfect random sequence. These are qualitatively grouped into terms such as "strong" or "weak" random numbers. If the random numbers are less than ideal or lack statistical randomness, the biases and dependencies in the bit stream can potentially compromise encrypted systems. Thus, it is imperative that the random numbers can be trusted to be theoretically random. In order to evaluate the statistical properties of random numbers, the bit sequences are submitted to various RNG tests, many of which are bundled into test suites such as NIST's Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications and Duke University's Dieharder Random Number Test Suite Bassham III *et al.* (2010); Brown *et al.* (2013). Most systems today use a pseudorandom number generator (PRNG), since they are easily integrated into electronic systems; however, true random number generators (TRNGs) provide fundamental advantages since their numbers are truly "random" in addition to statistical randomness.

PRNGs are exceptional choices for producing statistically random numbers quickly, even though they lack true randomness. These pseudorandom bit sequences are produced using various algorithms, which range in both computational requirements and complexity Akhshani *et al.* (2014); Han and Kim (2017); Li *et al.* (2010). Because there are no physical processes limiting these algorithms, the implementation of the algorithm can be made as fast as possible, and pseudorandom numbers can be made when needed without regard to lack of supply. However, the fact that these RNG schemes are entirely software based presents inherent weaknesses to the strength of these schemes. First, an exact replica of the scheme can be copied across many systems. For this reason, if a portion of the sequence is known, the rest of the sequence

Manuscript received: 11 May 2022,
Revised: 4 June 2022,
Accepted: 5 June 2022.

[1] rch0012@auburn.edu
[2] bkr0001@auburn.edu
[3] anr0025@auburn.edu
[4] deanron@auburn.edu
[5] edmonperkins@ncsu.edu (**Corresponding Author**)

can be extrapolated. Second, the initial state of the system (e.g., the "seed") exactly determines all future outputs of the system. Hence, the true randomness of PRNGs is extremely low despite their adoption throughout electronic, security, and communication systems.

Rather than producing statistically random numbers that can be replicated at will, true random number generators instead are based on a physical process (here, a chaotic circuit) that generates entropy. After the chaotic circuit generates entropy, a method is used to extract a bit of information from the dynamic response. Together, the chaotic system and the method of bit extraction may be considered a TRNG. One method for quantifying the entropy in a system of interest is in terms of a maximum Lyapunov exponent (MLE). This is the rate of divergence of two trajectories with almost identical initial conditions that are allowed to propagate in time. The maximum Lyapunov exponent also determines the maximum rate at which truly random bits can be extracted from the system Wolf *et al.* (1985). In essence, new information is available from the system at a specified rate, and once that information is extracted, there is a period of time before new information is available to be extracted again. Thus, a system with a large MLE gets this new information, and thus random bits, faster than a system with a small MLE.

Characteristics such as sensitivity to initial conditions, aperiodicity, and spread spectrum power density make chaotic systems ideal candidates for true random number generation Guinee and Blaszczyk (2009); Ergun and Ozoguz (2007); Pareschi *et al.* (2009); Blaszczyk and Guinee (2008). Chaotic systems that can be represented by sets of differential equations have the ability to be quantified as potential RNGs both from the ideal equations and from the implementation in hardware Valtierra *et al.* (2017); Sprott (2000); Tavas *et al.* (2010); Saito and Fujita (1981). Unfortunately, most chaotic systems that can be easily described with differential equations often do not lend themselves to simple electronic circuit implementation, while processes that are seemingly chaotic can be difficult to quantify accurately without prior knowledge of the underlying mechanics of the system.

Since the final goal of a TRNG is to get bits from a physical hardware process into a digital system, a scheme for forming these random bits from a process must be chosen. Many methods to achieve this are available, including sampling the process with an analog-to-digital converter, observing resulting clock jitter, and multiple oscillator sampling Cicek *et al.* (2014). Although the calculation of a maximum Lyapunov exponent in a system sets the maximum rate at which bits can be extracted from the system, there is no information determined about which bit sampling method to use. Often, the bit sampling method will necessitate using various post processing techniques to correct for the biases inherent in most sampling techniques of these physical systems. This is done to ensure the statistical randomness needed in order to pass the stringent testing that is required for random number generators Pareschi *et al.* (2010).

THE IDEAL JERK CHAOTIC SYSTEM

Many third order differential equations that exhibit chaotic behavior have previously been explored by Sprott Sprott (2010). These systems are known as "jerk" systems, because of their dependence on the third derivative with respect to time. Jerk systems have been implemented as a Josephson junction circuit Yalçin (2007), a diode-based circuit design Njitacke *et al.* (2017), and a smoothly-adjustable nonlinearity circuit Kengne *et al.* (2019).

This work focuses on a jerk oscillator that has a nonlinear term that is easily implemented in electronics: a signum function. Specifically, each of the integration stages can be implemented with operational amplifiers, and the signum function can be implemented as a high speed comparator Harrison *et al.* (2016). The jerk system of interest is given below as a third order differential equation in (1) and (2).

$$\dddot{x} = -0.5\ddot{x} - \dot{x} - x + sgn(x) \tag{1}$$

where

$$sgn(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases} \tag{2}$$

The nonlinearity in this system is caused by the signum function. The nonlinear dynamics for this type of system was well-defined by Sprott Sprott (2010). A modern implementation of this circuit using a comparator instead of a saturated operational amplifier is used in the current paper. A phase space plot of this system is shown in Fig. 1.

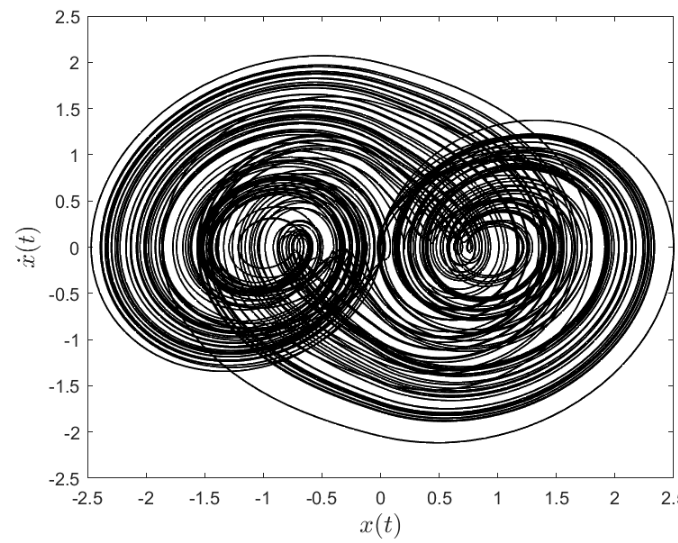


Figure 1 The ideal jerk equation's simulated phase space.

Next, the maximum Lyapunov exponent of the system needs to be estimated so that the maximum bit rate for random number generation can be found. The method chosen to accomplish this is a direct measurement of the divergence rate for many pairs of simulated trajectories that have almost identical initial conditions. The sensitivity of these chaotic systems to initial conditions causes trajectories that are different only by an amount well below measurement thresholds of real systems to quickly diverge. The MLE is then calculated by comparing the inital offset between the two systems with the time it takes for the difference in trajectories to reach a chosen threshold. This calculation is given with the following equation:

$$MLE = \frac{ln(\frac{threshold}{offset})}{time} \tag{3}$$

where *threshold* is the chosen divergence limit, *offset* is the initial difference in states of the two trajectories, and *time* is the final time taken to reach the threshold. Then, this translates into a theoretical maximum bit rate as follows:

$$bitrate = \frac{MLE}{ln\,2} \approx 1.443 * MLE \tag{4}$$

The bit rate given in eq. (4) has units of s^{-1}.

For the jerk system, 1000 time domain simulations are performed in order to determine the MLE by using a MATLAB script to implement the differential equations with a fixed time step. Initial conditions for both systems (the x, $\dot{x}$, and $\ddot{x}$ states) are randomized such that they were between -1 and $+1$, so that the trajectories remain in the chaotic region of the attractor rather than becoming globally unstable. Then, a very small offset (between 10^{-12} and 10^{-8}) is applied to the second system's x state. The threshold limit is chosen to be between 10^{-4} and 10^{-1}. The systems are then simulated forward in time until they reach the specified threshold. After this point, the systems diverge quickly.

The resulting MLE from many simulations with various initial conditions, thresholds, and offsets are between 0.152 and 0.153 for the jerk system. The MLE calculated in Sprott (2000) for the same system when using another method with some approximations is similar to this value. This then gives a bit rate of 0.218 and 0.221 bits per second. Examination of the system reveals that the system has a natural "pseudo frequency" of oscillation (when considering the time to complete one orbit around half of the attractor) of approximately 0.2 Hz. Thus, the bit rate of the system itself is approximately 1 bit per cycle. By framing the bit rate this way, the system can be scaled to any frequency, and the bit rate will remain constant with respect to the system. Specifically, when the system is implemented at high frequency in an electronic circuit, the circuit will be able to generate random bits at this higher natural frequency, as long as the system is represented accurately.

RANDOM NUMBER GENERATION

In order to obtain random bits from the system, the system is sampled at a fixed rate using an analog-to-digital converter (ADC). There are a number of different parameters for an ADC that can be chosen in regards to sampling, including voltage range, sampling frequency, and bits of resolution. In order for the jerk system to be a true random number generator, the sampling parameters of the ADC, which samples the system to produce bits that are statistically random, must be determined. These parameters are then replicated in hardware, and the physical electronic circuit is sampled in order to get truly random bits.

Simulation of an ADC sampling the system is achieved in MATLAB by implementing the jerk system equation. The 32-bit floating point value for x (since the x variable is sampled in hardware) is then converted to an n-bit sample value based on the chosen resolution of n bits and the ADC's maximum voltage. Successive samples are taken at approximately the natural pseudo frequency of the jerk system (i.e., at 0.2 Hz). For each sample, every bit of the sample except the lowest bit is discarded, and the lowest remaining bits are concatenated to form 8-bit random bytes.

The Dieharder test suite is used to evaluate the bit sequences generated from this simulation. There are 114 tests of randomness in this suite, but some tests are simply variations of other tests. However, in evaluating the sequences for randomness, the 114 tests are viewed as independent. Each test returns a P-value between 0 and 1, which is interpreted as follows: a P-value that is between 0.005 and 0.995 is considered to have passed that test, and a P-value of exactly 0 or 1 is considered to have failed. P-values that are under 0.005 and above 0.995 are "weak" and can be further resolved to either pass or fail through more testing. Due to the amount of data that Dieharder requires, some sequences that are actually random will produce weak P-values in approximately 1% of tests. An example output of Dieharder is shown in Fig. 2.

```
#=============================================================================#
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown
#=============================================================================#
   rng_name    |           filename             |rands/second|
 file_input_raw|           hard_3p125MHz_B16.bin|  1.92e+07  |
#=============================================================================#
        test_name   |ntup| tsamples |psamples|  p-value |Assessment
#=============================================================================#
   diehard_birthdays|   0|       100|     100|0.97980759|  PASSED
      diehard_operm5|   0|   1000000|     100|0.65329807|  PASSED
  diehard_rank_32x32|   0|     40000|     100|0.84539595|  PASSED
    diehard_rank_6x8|   0|    100000|     100|0.16086332|  PASSED
   diehard_bitstream|   0|   2097152|     100|0.52860159|  PASSED
        diehard_opso|   0|   2097152|     100|0.53328270|  PASSED
```

Figure 2 Partial output of Dieharder testing. More tests and results are given than are shown here.

The P-values returned by the Dieharder tests are such that if the tested bit sequence is statistically random, the P-values are uniformly distributed from 0 to 1. This allows for both individual test results and the results from the Dieharder suite as a whole to be analyzed quickly. This can be visually represented by plotting the test results versus a uniform distribution, in order to see the agreement. An example of this process is shown in Fig. 3 for simulation and hardware results. The P-values are sorted in ascending order (the type of test for each P-value is not taken into account) when plotted as a cumulative frequency against a straight line (the uniform distribution). Although this is only simulated data, it provides a baseline from which to build a hardware system that closely matches the parameters from the simulation equation and the analog-to-digital converter. From this testing, it is discovered that bit sequences gathered from ADC resolutions below 10-bits do not pass the Dieharder suite, indicating that these sequences are not statistically random. Above the 10-bit resolution mark, the P-values from the generated bit sequences are close to the desired uniform distributions.

These simulation test results are performed by sampling at 1 bit per cycle, and almost ideal results are obtained at the 12-bit resolution level. When the sampling frequency is increased to 2.5 bits and 5 bits per cycle at 12-bits of resolution, the bit sequences still pass most or all of the Dieharder test suite. These plots indicate that statistically random bits can be obtained from a system that is sampled faster than the maximum Lyapunov exponent dictates for true random number generation. Thus, it is imperative that the implementation of the random number generation take into account the theoretical limitation for true randomness. Beyond this limit, the system as a whole cannot be truly random, even though statistical randomness might be achieved at a higher bit rate.

HARDWARE CIRCUIT DESIGN AND TESTING

The circuit presented in this paper, which is shown in Fig. 4, includes additional components to make testing easier. Preliminary simulations of this circuit were presented in Harrison *et al.* (2019), and a preliminary design of this circuit was described in Harrison *et al.* (2016) and Harrison *et al.* (2017). Specifically, the current circuit has pin headers used to power the board that are exchanged with a micro USB connector to enable power to come directly from a number of readily available power sources, including a host computer with an open USB port or a mobile battery bank for testing inside of an enclosed space. A PMOS transistor is placed after this connector in order to apply power to the rest of the circuit in a more controlled fashion (via unshorting a jumper on the board

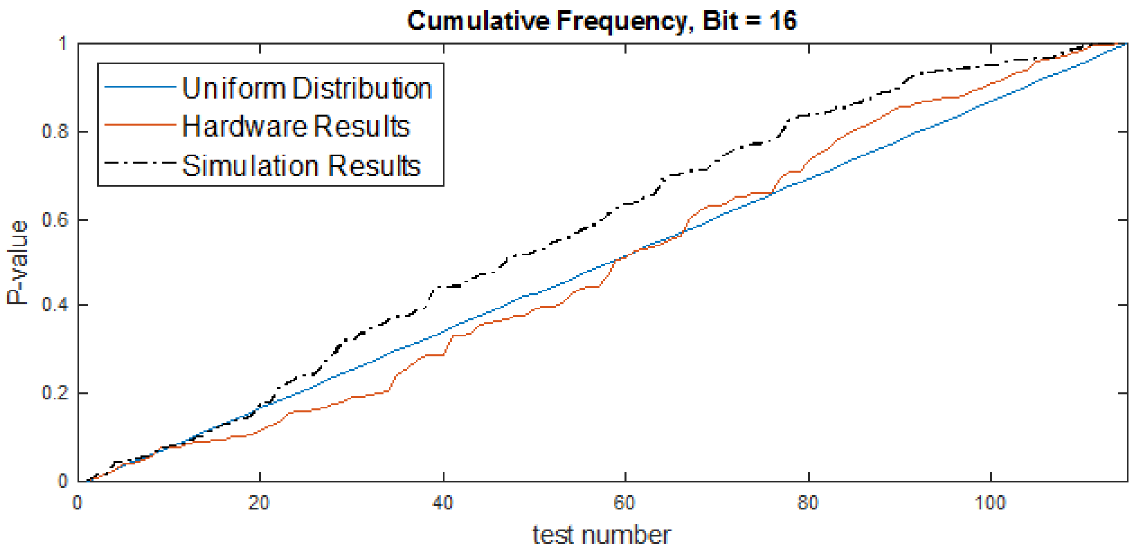


Figure 3 Cumulative frequency plot of Dieharder test results from simulating the 16^{th} bit of an ADC in software, and hardware implementation of ADC sampling of the jerk circuit at 3.125 MSPS.

from ground) than just plugging in the connector. Also, in testing previous versions of the board, it was discovered that the circuit could enter a periodic railing state upon being powered, but it could then be forced into the normal chaotic state by temporarily shorting the x node of the op amp integrators to ground.

A small momentary switch is added to this node to facilitate this correction. Finally, the signum output is taken from the $-Q$ pin on the comparator so as to not interfere with the Q signal in the feedback path. The Q pin is the signum output of the comparator, which is directly connected to the rest of the feedback summing circuitry. In earlier versions of the board, probing this pin directly caused some undesired changes to the chaotic attractor, since the oscilloscope loaded the pin. Essentially, if probed, the signum output would no longer be accurate. To remedy this, the chip has a $-Q$ output that can be probed instead, which can be inverted if needed. By probing the $-Q$ pin, this leaves the Q output undisturbed. As another solution, the Q output could instead have been buffered with another chip (e.g., an operational amplifier), but the current board was designed with a minimal part count in mind. The pseudo-fundamental frequency of the board was maintained at 4 MHz. A picture of the front and back of the updated board is shown in Fig. 5.

It should be noted that the 15kΩ resistors are a deviation from the ideal operational amplifier integration circuit, and, in fact, these convert the ideal integrators into active amplifier circuits with a low pass filter. Since the gain of these integrators was high, the operational amplifier integrators could easily saturate on startup, which would prevent oscillations from occurring. The 15kΩ resistors prevent saturation of the feedback capacitors. The 15kΩ value for these resistors was found from trial and error: a resistance that is too high causes the capacitor to saturate, while a resistance that is too low causes the oscillations to be considerably damped (e.g., the chaotic signal would stay on each side of the attractor for too long before switching, which reduces the Lyapunov exponent of the implemented circuit). This value was chosen for good oscillation characteristics compared with the ideal circuit.

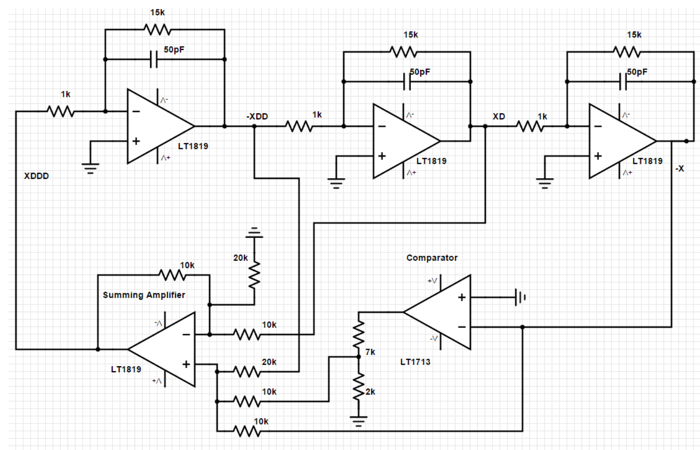


Figure 4 A schematic of the electronic implementation of the jerk oscillator.

For testing of this circuit as a true random number generator, ADC sampling is achieved using a Handyscope HS6 USB Oscilloscope from TiePie Engineering connected to a host computer running the provided MultiChannel software. The circuit is powered from a USB port on the same computer. The HS6 allows for up to 16 bit streaming ADC sampling, but at that resolution the sampling speed is limited to 3.125 MS/s, slightly less than the desired 4 MS/s to achieve a 1 bit per cycle random output. The full scale voltage of the ADC is chosen to be ± 2 V since the circuit has peaks of approximately ± 1 V when AC coupled to the oscilloscope and powered using the single 5V supply from the USB port. This results in a loss of approximately 1 bit of resolution when compared to a full scale voltage that is smaller, but the next lowest supported by the MultiChannel software is 800 mV. A screenshot of the circuit being sampled in this software is shown in Fig. 6.

32 GB of data is collected from the x node of the circuit at 3.125 MS/s and then split using a MATLAB script into sixteen 2 GB files, one with each separate bit of the 16-bit samples concatenated together. These files are then subjected to Dieharder testing in the same manner as the jerk equation simulation data. These results are shown in Fig. 3.

For the hardware circuit, the bit sequences pass most of the Dieharder tests starting at the 13^{th} bit and do not fail any tests at the 16^{th} bit of the ADC sample. At the 12^{th} bit and above, the sequences systematically fail certain sets of tests, most notably the sts_serial and rgb_lagged_sum series of tests. These tests involve skipping many bits in a row and thus the input file is rewound multiple times for each of these tests. This can potentially make the bit sequence seem like it is repeating itself, but this is unavoidable with this setup of Dieharder without a much larger input file size.

A concern for this method of sampling (i.e., taking one bit per sample of a high resolution sample) is that the lowest bits are masked by noise in the system, and thus the randomness ultimately achieved may be due to noise and not to the chaotic dynamics in the system. Although the noise floor is a useful metric for linear systems, noise and nonlinearity can produce unintuitive dynamics. For instance, noise can cause a system undergoing chaos to become regular Lepik and Hein (2005), but it can also drive a system undergoing regular motion to become chaotic Perkins and Balachandran (2012). Further, noise can cause stochastic resonance Perkins and Balachandran (2015), modify the hysteresis curve of nonlinear oscillators Perkins (2017); Perkins and Fitzgerald (2018), affect the dynamics of intrinsic localized modes in coupled oscillator arrays Perkins *et al.* (2016); Balachandran *et al.* (2015), and cause learning to be degraded in an adaptive oscillator circuit Li *et al.* (2021).

In order to investigate whether the noise had an effect on the randomness of the jerk oscillator, two additional tests are performed using Dieharder. In the first test, the +5V power rail on the board is used to generate random bits using the same sampling parameters at the 16^{th} bit of resolution. In the second test, a separate chaotic circuit with higher fundamental frequency is likewise sampled. Both of these bit sequences fail the majority of the Dieharder suite with 2 GB input files. If the sampling or thermal noise in the jerk chaos board was providing the randomness in order to pass Dieharder, then the sequences generated from the noise should pass the test suite. These results show that the statistical randomness achieved using this particular nonlinear circuit and bit extraction technique is likely coming from the nonlinear dynamics, instead of from electronic noise.

Although there is no real way to prove randomness, the results from the Dieharder RNG test suite indicate that the jerk circuit is able to provide statistically random bits from a hardware source under various circumstances. This test suite is widely used to stringently test pseudorandom number generators with much larger bit sequences available on demand. Since the chaotic jerk circuit is implemented in a small form factor with commercial off the shelf components, it can easily be integrated into other systems requiring truly random bits at high speeds. The test results from both the simulated and hardware TRNG are in good agreement with each other. Overall, the jerk oscillator circuit is an ideal candidate for random number generation.

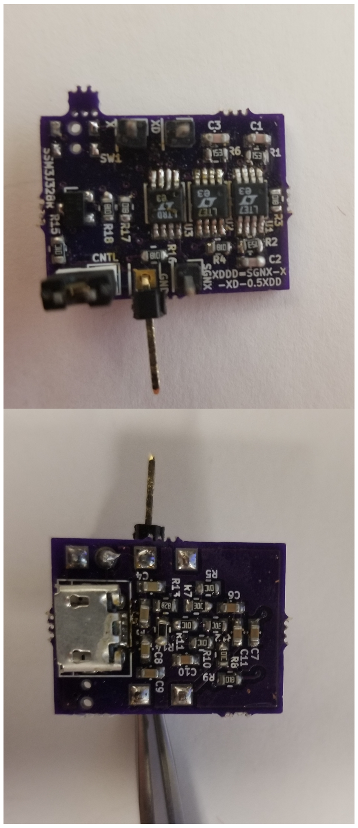


Figure 5 The front and back of the populated circuit board that implements the jerk equation at 4 MHz.

Figure 6 Data readout from the Handyscope HS6 from the hardware circuit.

CONCLUSION

A scheme for extracting true random numbers directly from a chaotic jerk system is shown. This system is implemented in high speed electronics on a small printed circuit board and sampled in accordance with the necessary parameters found from the simulation results. The bit sequences generated from the physical system pass the Dieharder Random Number Generator test suite, which enables this system to function as a fast random bit generator for many different applications. Overall, this system shows high dynamic complexity in a compact form, which is desirable for a true random number generator.

Acknowledgements

Support for this project under grant number **G00009556/7** from the U.S. Army (research sponsor) and Torch Technologies, Inc. (prime contractor) is greatly appreciated.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Availability of data and material

Not applicable.

LITERATURE CITED

Akhshani, A., A. Akhavan, A. Mobaraki, S.-C. Lim, and Z. Hassan, 2014 Pseudo random number generator based on quantum chaotic map. Communications in Nonlinear Science and Numerical Simulation **19**: 101–111.

Balachandran, B., E. Perkins, and T. Fitzgerald, 2015 Response localization in micro-scale oscillator arrays: influence of cubic coupling nonlinearities. International Journal of Dynamics and Control **3**: 183–188.

Bassham III, L. E., A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, *et al.*, 2010 *Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications*. National Institute of Standards & Technology.

Blaszczyk, M. and R. A. Guinee, 2008 A true random binary sequence generator based on chaotic circuit. In *IET Irish Signals and Systems Conference (ISSC 2008)*, pp. 294–299.

Brown, R. G., D. Eddelbuettel, and D. Bauer, 2013 Dieharder: A random number test suite. Open Source software library, under development, URL http://www. phy. duke. edu/˜ rgb/General/dieharder. php .

Cicek, I., A. E. Pusane, and G. Dundar, 2014 A novel design method for discrete time chaos based true random number generators. INTEGRATION, the VLSI journal **47**: 38–47.

Ergun, S. and S. Ozoguz, 2007 A chaos-modulated dual oscillator-based truly random number generator. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pp. 2482–2485, IEEE.

Guinee, R. A. and M. Blaszczyk, 2009 A novel true random binary sequence generator based on a chaotic double scroll oscillator combination with a pseudo random generator for cryptographic applications. In *Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for*, pp. 1–6, IEEE.

Han, M. and Y. Kim, 2017 Unpredictable 16 bits LFSR-based true random number generator. In *SoC Design Conference (ISOCC), 2017 International*, pp. 284–285, IEEE.

Harrison, R. C., B. K. Rhea, A. N. Ramsey, R. N. Dean, and J. E. Perkins, 2019 A true random number generator based on a chaotic jerk system. In *2019 SoutheastCon*, pp. 1–5, IEEE.

Harrison, R. C., B. K. Rhea, F. T. Werner, and R. Dean, 2017 A compact and low power realization of a high frequency chaotic oscillator. Additional Conferences (Device Packaging, HiTEC, HiTEN, & CICMT) **2017**: 4.

Harrison, R. C., B. K. Rhea, F. T. Werner, and R. N. Dean, 2016 A 4 MHz chaotic oscillator based on a jerk system. In *International Conference on Applications in Nonlinear Dynamics*, pp. 41–51, Springer.

Kengne, J., R. L. T. Mogue, T. F. Fozin, and A. N. K. Telem, 2019 Effects of symmetric and asymmetric nonlinearity on the dynamics of a novel chaotic jerk circuit: Coexisting multiple attractors, period doubling reversals, crisis, and offset boosting. Chaos, Solitons & Fractals **121**: 63–84.

Lepik, Ü. and H. Hein, 2005 On response of nonlinear oscillators with random frequency of excitation. Journal of sound and vibration **288**: 275–292.

Li, C.-Y., T.-Y. Chang, and C.-C. Huang, 2010 A nonlinear PRNG using digitized logistic map with self-reseeding method. In *VLSI Design Automation and Test (VLSI-DAT), 2010 International Symposium on*, pp. 108–111, IEEE.

Li, X., M. R. E. U. Shougat, T. Mollik, A. N. Beal, R. N. Dean, *et al.*, 2021 Stochastic effects on a hopf adaptive frequency oscillator. Journal of Applied Physics **129**: 224901.

Njitacke, Z., L. K. Kengne, *et al.*, 2017 Antimonotonicity, chaos and multiple coexisting attractors in a simple hybrid diode-based jerk circuit. Chaos, Solitons & Fractals **105**: 77–91.

Pareschi, F., G. Scotti, L. Giancane, R. Rovatti, G. Setti, *et al.*, 2009 Power analysis of a chaos-based random number generator for cryptographic security. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pp. 2858–2861, IEEE.

Pareschi, F., G. Setti, and R. Rovatti, 2010 Statistical testing of a chaos based CMOS true-random number generator. Journal of Circuits, Systems, and Computers **19**: 897–910.

Perkins, E., 2017 Effects of noise on the frequency response of the monostable duffing oscillator. Physics Letters A **381**: 1009–1013.

Perkins, E. and B. Balachandran, 2012 Noise-enhanced response of nonlinear oscillators. Procedia Iutam **5**: 59–68.

Perkins, E. and B. Balachandran, 2015 Effects of phase lag on the information rate of a bistable duffing oscillator. Physics Letters A **379**: 308–313.

Perkins, E. and T. Fitzgerald, 2018 Continuation method on cumulant neglect equations. Journal of Computational and Nonlinear Dynamics **13**.

Perkins, E., M. Kimura, T. Hikihara, and B. Balachandran, 2016 Effects of noise on symmetric intrinsic localized modes. Nonlinear Dynamics **85**: 333–341.

Saito, T. and H. Fujita, 1981 Chaos in a manifold piecewise linear system. Electronics and Communications in Japan (Part I: Communications) **64**: 9–17.

Sprott, J. C., 2000 Simple chaotic systems and circuits. American Journal of Physics **68**: 758–763.

Sprott, J. C., 2010 *Elegant chaos: algebraically simple chaotic flows*. World Scientific.

Sundaresan, S., R. Doss, S. Piramuthu, and W. Zhou, 2015 Secure tag search in RFID systems using mobile readers. IEEE Transactions on Dependable and Secure Computing **12**: 230–242.

Tavas, V., A. S. Demirkol, S. Ozoguz, A. Zeki, and A. Toker, 2010 An ADC based random bit generator based on a double scroll chaotic circuit. Journal of Circuits, Systems, and Computers **19**: 1621–1639.

Valtierra, J. L., E. Tlelo-Cuautle, and Á. Rodríguez-Vázquez, 2017 A switched-capacitor skew-tent map implementation for random number generation. International Journal of Circuit Theory and Applications **45**: 305–315.

Volos, C. K., 2013 Image encryption scheme based on coupled chaotic systems. Journal of Applied Mathematics and Bioinformatics **3**: 123.

Wolf, A., J. B. Swift, H. L. Swinney, and J. A. Vastano, 1985 Determining lyapunov exponents from a time series. Physica D: Nonlinear Phenomena **16**: 285–317.

Yalçin, M. E., 2007 Multi-scroll and hypercube attractors from a general jerk circuit using josephson junctions. Chaos, Solitons & Fractals **34**: 1659–1666.

How to cite this article: Harrison, R. C., Rhea, B. K., Oldag, A. R., Dean, R. N. and Perkins, E. Experimental Validation of a Chaotic Jerk Circuit Based True Random Number Generator. Chaos Theory and Applications, 4(2), 64-70, 2022.

CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.4 / No.2 / 2022 / pp.71-81
https://doi.org/10.51537/chaos.1096278

Stability and Hopf Bifurcation Analysis of a Fractional-order Leslie-Gower Prey-predator-parasite System with Delay

Xiaoting Yang [*,1], **Liguo Yuan** [α,*,2] **and Zhouchao Wei** [γ,3]

[*]Department of Mathematics, College of Mathematics and Informatics, South China Agricultural University, Guangzhou, 510642, P. R. China, [α]School of Mathematical Sciences, Anhui University, Hefei, 230601, P. R. China, [γ]School of Mathematics and Physics, China University of Geosciences, Wuhan, 430074, P. R. China.

ABSTRACT A fractional-order Leslie-Gower prey-predator-parasite system with delay is proposed in this article. The existence and uniqueness of the solutions, as well as their non-negativity and boundedness, are studied. Based on the characteristic equations and the conditions of stability and Hopf bifurcation, the local asymptotic stability of each equilibrium point and Hopf bifurcation of interior equilibrium point are investigated. Moreover, a Lyapunov function is constructed to prove the global asymptotic stability of the infection-free equilibrium point. Lastly, numerical examples are studied to verify the validity of the obtained newly results.

KEYWORDS
Fractional derivative
Hopf bifurcation
Stability
Leslie-Gower prey-predator-parasite system

INTRODUCTION

Ecosystem is an extremely complex dynamics system. Mathematicians have great interest in dynamical characteristics of ecosystem. Especially, ecology and epidemiology attract more and more mathematicians' attention. Although ecology and epidemiology are two different fields, they get closer and closer for years(Anderson and May 1980; Zhou *et al.* 2010; Mbava *et al.* 2017; Shaikh *et al.* 2018; Adak *et al.* 2020). In 1980, Anderson and May first to study the eco-epidemiological model with disease in the prey(Anderson and May 1980). Recently, Zhou et al considered a predator-prey model with modified Leslie-Gower functional response and studied the Hopf bifurcation of this model(Zhou *et al.* 2010). They found that when the rate of infection exceeds a critical value, the strictly positive interior equilibrium experiences Hopf bifurcation. The eco-epidemic predator-prey model exhibits interesting dynamics with infected predators. So, Shaikh et al considered the stability of a Holling type III response mechanism for predation(Shaikh *et al.* 2018). The predator faced enormous competition from super-predators and even faced extinction. The disease was regarded as a biological control that allowed predator populations to recover from low numbers. Hence, Mbava et al considered a predator-prey model with disease in super-predator and studied its dynamic properties(Mbava *et al.* 2017). In addition, Adak et al analyzed the chaos and Hopf bifurcation of the delay-induced Leslie-Gower predator-prey-parasite model(Adak *et al.* 2020). It can be seen that research on eco-epidemiological models is a hot topic.

Fractional calculus is an extension of classical calculus. In recent years, fractional calculus has developed rapidly, which has gradually penetrated into scientific and engineering application fields. Furthermore, it also has become an important tool in many fields(Kilbas *et al.* 2006; Rajagopal *et al.* 2020; Li and Chen 2004). Compared with integer-order derivative, the fractional derivative has better memory. It can excellently describe long-range temporal memory(Rihan and Rajivganthi 2020). Since most biological models have long-range temporal memory, it is significant to consider the fractional derivative into account. Currently, research on this area has some outstanding results(Yousef *et al.* 2021; Li *et al.* 2017a; Boukhouima *et al.* 2017; Moustafa *et al.* 2020). Yousef et al analyzed the influence of fear and fractional-order derivative on system dynamics(Yousef *et al.* 2021). Li et al investigated the stability of a fractional-order predator-prey model, which incorporates a prey refuge(Li *et al.* 2017a). Boukhouima et al studied a fractional-order model to describe the dynamics of human immunodeficiency virus infection(Boukhouima *et al.* 2017). Mousfata et al consider a fractional-order eco-epidemiological system of prey

Manuscript received: 31 March 2022,
Revised: 21 May 2022,
Accepted: 5 June 2022.

[1] 13437617328@163.com
[2] liguoy@scau.edu.cn
[3] weizhouchao@163.com (**Corresponding Author**)

population with disease. Moreover, the dynamics of this model was analyzed(Moustafa *et al.* 2020).

Delay plays an important part in ecosystem and it exists universally. Different models take different biological delays into account(Tao *et al.* 2018; Shi *et al.* 2022; Chinnathambi and Rihan 2018; Fernández-Carreón *et al.* 2022; Rihan and Rajivganthi 2020; Xu and Zhang 2013; Huang *et al.* 2019; Mahmoud *et al.* 2017; Pu 2020; Alidousti and Mostafavi Ghahfarokhi 2019; Huang *et al.* 2020; Deng *et al.* 2007; Kashkynbayev and Rihan 2021; Yuan *et al.* 2013). Compared with the systems without delay, the systems with delays will show more complex nonlinear dynamic behavior. Delay may cause the equilibrium points instability. Moreover, spreading of disease is not happen immediately. In general, infectious disease has an incubation period. Therefore, it is important to take delay into account for biological model, and it will describe real life more accurately.

In (Zhou *et al.* 2010), Zhou et al formulated the following system

$$\begin{aligned}
\frac{dS(t)}{dt} &= rS(t)(1-\frac{S(t)+I(t)}{K}) - \beta S(t)I(t),\\
\frac{dI(t)}{dt} &= \beta S(t)I(t) - cI(t) - \frac{c_1 I(t)y(t)}{I(t)+K_1},\\
\frac{dy(t)}{dt} &= y(t)(a_2 - \frac{c_2 y(t)}{I(t)+K_2}),
\end{aligned} \tag{1}$$

and studied the dynamics of (1). Based on the importance of delay, Adak et al considered delay into account and formulated the following system(Adak *et al.* 2020)

$$\begin{aligned}
\frac{dS(t)}{dt} &= rS(t)(1-\frac{S(t)+I(t)}{K}) - \beta S(t)I(t),\\
\frac{dI(t)}{dt} &= \beta S(t-\tau)I(t-\tau) - cI(t) - \frac{c_1 I(t)y(t)}{I(t)+K_1},\\
\frac{dy(t)}{dt} &= y(t)(a_2 - \frac{c_2 y(t)}{I(t)+K_2}).
\end{aligned} \tag{2}$$

They exhibited the dynamic behavior of system (2), such as chaos and Hopf bifurcation. However, Zhou and Adak et al did not take the good memory characteristics of fractional derivative into account, which can well describe long-range temporal memory. Hence, we consider the fractional derivative into account for system (2) and establish a fractional-order Leslie-Gower prey-predator-parasite system with delay

$$\begin{aligned}
D^\alpha S(t) &= rS(t)(1-\frac{S(t)+I(t)}{K}) - \beta S(t)I(t),\\
D^\alpha I(t) &= \beta S(t-\tau)I(t-\tau) - cI(t) - \frac{c_1 I(t)y(t)}{I(t)+K_1},\\
D^\alpha y(t) &= y(t)(a_2 - \frac{c_2 y(t)}{I(t)+K_2}).
\end{aligned} \tag{3}$$

The initial conditions of (3) are as follows:

$$S(t) = \eta_1(t), I(t) = \eta_2(t), y(t) = \eta_3(t), t \in [-\tau, 0], \tag{4}$$

where $S(t), I(t), y(t)$ represent the growth rates of susceptible prey, infected prey and predator population at time t respectively. r represents intrinsic growth rate of susceptible prey. K represents environmental prey carrying capacity. β represents infection rate. c represents predation-independent death rate of infectious prey. c_1 represents maximum predation rate of predator on an infectious prey. K_1 represents half-saturation density. a_2 represents intrinsic growth rate of predator. c_2 and K_2 are positive constants. D^α denotes α-order Caputo differential derivative, $\alpha \in (0,1]$, and $r, K, \beta, c, c_1, K_1, a_2, c_2, K_2$ are all nonnegetive. Label R^3_+ as the non-negative cone, $\eta = (\eta_1(t), \eta_2(t), \eta_3(t)) \in C([-\tau, 0], R^3_+)$, the Banach space of continuous real-valued functions on the interval $[-\tau, 0]$ with norm $||\eta|| = \sup_{-\tau \le t \le 0} |\eta(t)|$, and $\eta_1(t) \ge 0, \eta_2(t) \ge 0, \eta_3(t) \ge 0, \eta_1(0) > 0, \eta_2(0) > 0, \eta_3(0) > 0$.

We aim to investigate the stability of system (3) and how the delay affects the dynamics of this system. Firstly, we investigate the existence and uniqueness of the solutions, as well as their non-negativity and boundedness. Furthermore, we derive the local asymptotic stability of every equilibrium point. Then, we demonstrate the global asymptotic stability of the infection-free equilibrium point by formulating a Lyapunov function. Moreover, we choose delay as the bifurcation parameter to show interior equilibrium point occurs Hopf bifurcation under some conditions. Lastly, we give the numerical examples to back up our results.

The structure of this article is as follows. We describe basic concepts in section 2. The existence and uniqueness of the solutions, as well as their non-negativity and boundedness are investigated in section 3. Besides, we derive equilibrium points and the local asymptotic stability corresponding to each equilibrium point. Then, we analyze the Hopf bifurcation of the interior equilibrium point. We provide two illustrative examples to back up our findings in section 4. Finally, we close the paper in last section.

MATHEMATICAL PRELIMINARIES

Definition 1. (Kilbas *et al.* 2006) The Riemann-Liouville's fractional integral of order $\alpha > 0$ for a function f is defined as

$$D^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} f(s) ds, t > 0,$$

where $\Gamma(\cdot)$ is the Gamma function.

Definition 2. (Kilbas *et al.* 2006) The Caputo's fractional derivative of order α for a function f is defined as

$$D^\alpha f(t) = \frac{1}{\Gamma(m-\alpha)} \int_0^t (t-s)^{m-\alpha-1} f^{(m)}(s) ds, t > 0,$$

where $0 \le m-1 \le \alpha < m, m \in Z^+$.

.

Lemma 1. (Wang *et al.* 2011) Consider the following nonlinear differential equation with Caputo fractional derivative

$$\begin{aligned}
D^\alpha X(t) &= f(X(t)) + g(X(t-\tau)),\\
X(t) &= \Phi(t), t \in [-\tau, 0],
\end{aligned} \tag{5}$$

where $\alpha \in (0,1], X(t) \in R^n, \tau \ge 0$, then the characteristic equation of system is

$$|s^\alpha E - A - Be^{-s\tau}| = 0,$$

where A and B is the Jacobian matrix of the function $f(X(t))$ and $g(X(t))$ at the equilibrium point of the system (5). The zero solution of system (5) is locally asymptotically stable if all the roots of the characteristic equation restricted to $arg(\lambda) > \frac{\pi\alpha}{2}$ have negative real parts.

Lemma 2. (Odibat and Shawagfeh 2007) Suppose that $f(t) \in C[a,b]$ and $D^\alpha f(t) \in C[a,b]$ for $0 < \alpha \le 1$. If $D^\alpha f(t) \ge 0, \forall t \in [a,b]$, then $f(t)$ is non-decreasing for each $t \in [a,b]$. If $D^\alpha f(t) \le 0, \forall t \in (a,b)$, then $f(t)$ is non-increasing for each $t \in [a,b]$.

Lemma 3. (Li *et al.* 2010) Consider the system

$$D^{\alpha}x(t) = f(t,x), t > t_0, \tag{6}$$

with initial condition $x(t_0)$, where $\alpha \in (0,1], f : [t_0,\infty) \times \Omega \to R^n, \Omega \subseteq R^n$, if $f(t,x)$ satisfies the locally lipschitz condition with respect to x, then there exists a unique solution of (6) on $[t_0,\infty) \times \Omega$.

Lemma 4. (Cruz 2015) Let $x(t) \in R^+$ be a continuous and derivative function. Then, for any time instant $t \geq t_0$,

$${}_{t_0}D_t^{\alpha}(x(t) - x^* - x^* \ln \frac{x(t)}{x^*}) \leq (1 - \frac{x^*}{x(t)}){}_{t_0}D_t^{\alpha}x(t), \tag{7}$$

where $\forall\alpha \in (0,1), x^* \in R^+$.

Lemma 5. (Li *et al.* 2017b) Let $u(t) \in C([0,+\infty))$. If $u(t)$ satisfies $D^{\alpha}u(t) \leq a - bu(t)$, $u(0) = u_0$, where $\alpha \in (0,1], (a,b) \in R^2$ and $b \neq 0$, then

$$u(t) \leq (u_0 - \frac{a}{b})E_{\alpha}(-bt^{\alpha}) + \frac{a}{b}.$$

MAIN RESULTS

Existence and Uniqueness of solutions

Theorem 6. For any non-negative initial conditions the fractional-order system (3) has a unique solution.

Proof. Consider the region $\Pi = \{(S,I,y) \in R^3 : \max\{|S|,|I|,|y|\} \leq M\}$, and denote $X = (S,I,y)$, $\widehat{X} = (\widehat{S},\widehat{I},\widehat{y})$, then define a mapping $f(X) = (f_1(X), f_2(X), f_3(X))$, where

$$\begin{aligned} f_1(X) &= rS(t)(1 - \frac{S(t)+I(t)}{K}) - \beta S(t)I(t), \\ f_2(X) &= \beta S(t-\tau)I(t-\tau) - cI(t) - \frac{c_1 I(t)y(t)}{I(t)+K_1}, \\ f_3(X) &= y(t)(a_2 - \frac{c_2 y(t)}{I(t)+K_2}). \end{aligned}$$

For $X, \widehat{X} \in \Pi$, then

$$\begin{aligned} &\|f(X) - f(\widehat{X})\| = |f_1(X) - f_1(\widehat{X})| + |f_2(X) - f_2(\widehat{X})| \\ &+ |f_3(X) - f_3(\widehat{X})| \\ &= |r(S-\widehat{S}) - \frac{r}{K}(S^2 - \widehat{S}^2) - (\frac{r}{K} + \beta)(SI - \widehat{S}\widehat{I})| \\ &+ |\beta I(t-\tau)(S(t-\tau) - \widehat{S}(t-\tau)) + \beta\widehat{S}(t-\tau)\times \\ &(I(t-\tau) - \widehat{I}(t-\tau)) - c(I-\widehat{I}) - \frac{c_1 I\widehat{I}(y-\widehat{y})}{(I+K_1)(\widehat{I}+K_1)} \\ &- \frac{c_1K_1I(y-\widehat{y}) + c_1K_1\widehat{y}(I-\widehat{I})}{(I+K_1)(\widehat{I}+K_1)}| + |a_2(y-\widehat{y}) \\ &- \frac{K_2c_2(y+\widehat{y})(y-\widehat{y}) + c_2I(y^2-\widehat{y}^2) - c_2y^2(I-\widehat{I})}{(I+K_2)(\widehat{I}+K_2)}| \\ &\leq (r + 3\frac{Mr}{K} + \beta M)|S-\widehat{S}| + M(\frac{r}{K}+\beta)|I-\widehat{I}| \\ &+ \beta M|S(t-\tau) - \widehat{S}(t-\tau)| + \beta M|I(t-\tau) - \widehat{I}(t-\tau)| \\ &+ (c + \frac{c_1K_1M}{K_1^2})|I-\widehat{I}| \\ &+ (\frac{c_1M^2 + c_1K_1M}{K_1^2} + a_2 + \frac{2MK_2c_2 + 2M^2c_2}{K_2^2})|y-\widehat{y}| \\ &+ \frac{c_2M^2}{K_2^2}|I-\widehat{I}| \\ &= (r + 3\frac{Mr}{K} + 2\beta M)|S-\widehat{S}| + (M\frac{r}{K} + 2\beta M + c \\ &+ \frac{c_1K_1M}{K_1^2} + \frac{c_2M^2}{K_2^2})|I-\widehat{I}| + (\frac{c_1M^2 + c_1K_1M}{K_1^2} + a_2 \\ &+ \frac{2MK_2c_2 + 2M^2c_2}{K_2^2})|y-\widehat{y}| \\ &\leq L\|X - \widehat{X}\|, \end{aligned}$$

where $L = \max\{(r + 3\frac{Mr}{K} + 2\beta M), (\frac{Mr}{K} + 2\beta M + c + \frac{c_1K_1M}{K_1^2} + \frac{c_2M^2}{K_2^2}), (\frac{c_1M^2+c_1K_1M}{K_1^2} + a_2 + \frac{2MK_2c_2+2M^2c_2}{K_2^2})\}$. Hence, Lipschitz condition is satisfied for $f(X)$. There exist a unique solution of system (3) on the basis of Lemma 3. □

Non-negativity of solutions

Theorem 7. All the solutions of system (3) starting from

$$D_+ = \{(S,I,y) \in R^3 : S,I,y \in R^+\},$$

are non-negative.

Proof. Above all, we derive that the solution $S(t)$ starting from D_+ is non-negative, i.e. $S(t) \geq 0$ for $t \geq t_0$. Suppose that is not true, then there exist $t_1 > t_0$ such that $S(t) > 0, t_0 \leq t < t_1, S(t_1) = 0, S({t_1}^+) < 0$. From the first equation of system (3), we get

$$D^{\alpha}S(t_1)|_{S(t_1)=0} = 0.$$

Based on the Lemma 2, there exsits $S({t_1}^+) = 0$ and it contradicts with $S({t_1}^+) < 0$. Hence, we can get $S(t) \geq 0$ for $t \geq t_0$.

If there exist $t_2 > t_0$ such that $I(t) > 0, t_0 \leq t < t_2, I(t_2) = 0, I({t_2}^+) < 0$, then we get

$$D^{\alpha}I(t_2)|_{I(t_2)=0} = \beta S(t_2-\tau)I(t_2-\tau) > 0.$$

Based on the Lemma 2, there exsits $I({t_2}^+) > 0$ and it contradicts with $I({t_2}^+) < 0$. So, we get $I(t) \geq 0$ for $t \geq t_0$.

If there exist a constant $t_3 > t_0$ such that $y(t) > 0, t_0 \leq t < t_3, y(t_3) = 0, y(t_3{}^+) < 0$, then we get

$$D^\alpha y(t_3)|_{y(t_3)=0} = 0.$$

Similarly, we have $y(t_3{}^+) = 0$, which contradicts with $y(t_3{}^+) < 0$. Hence, we obtain $y(t) \geq 0$ for $t \geq t_0$. □

Boundedness of solutions

Theorem 8. All solutions of system (3) starting from R^3_+ are bounded.

Proof. Denote

$$f(S(t)) = rS(t)(1 - \frac{S(t) + I(t)}{K}) - \beta S(t)I(t),$$
$$F(S(t)) = rS(t)(1 - \frac{S(t)}{K}),$$

and let

$$D^\alpha S(t) = f(S(t)), \tag{8}$$
$$D^\alpha S(t) = F(S(t)). \tag{9}$$

Assume $h(t)$ is the solution of (8) and $H(t)$ is the solution of (9). Since $f(S(t)) \leq F(S(t))$, we can derive $h(t) \leq H(t)$ according to the comparison theorems of fractional-order differential equations(Hu *et al.* 2009). Let $z_1(t) = \frac{rS(t)}{K}$, then (9) become

$$D^\alpha z_1(t) = z_1(t)(r - z_1(t)). \tag{10}$$

Denote $\bar{H}(t)$ is the solution of (10), then $\bar{H}(t) = \frac{rH(t)}{K}$. Based on the methods in (Li *et al.* 2019), we can get $\limsup_{t\to\infty} z_1(t) \leq \hat{m}$, thus we can derive $\limsup_{t\to\infty} S(t) \leq \frac{K\hat{m}}{r}$, denote $m = \frac{K\hat{m}}{r}$, then $\limsup_{t\to\infty} S(t) \leq m$. Define a function $W(t) = S(t-\tau) + I(t)$. Then

$$\begin{aligned} D^\alpha W(t) &= D^\alpha S(t-\tau) + D^\alpha I(t) \\ &= rS(t-\tau)(1 - \frac{S(t-\tau) + I(t-\tau)}{K}) \\ &\quad - cI(t) - \frac{c_1 I(t)y(t)}{I(t) + K_1} \\ &\leq rS(t-\tau) - cI(t) \\ &= 2rS(t-\tau) - dW(t) \\ &\leq 2rm - dW(t), \end{aligned}$$

where $d = \min\{r, c\}$. From Lemma 5, we can get

$$0 \leq W(t) \leq (W(0) - \frac{2rm}{d})E_\alpha(-dt^\alpha) + \frac{2rm}{d},$$

where E_α is the Mittag-Leffler function. Hence, we can obtain $\limsup_{t\to\infty} W(t) \leq \frac{2rm}{d}$. Then $\limsup_{t\to\infty} I(t) \leq \frac{2rm}{d}$. For the third equation of system (3), we can obtain

$$D^\alpha y(t) \leq y(t)(a_2 - \frac{dc_2 y(t)}{2rm + dK_2}). \tag{11}$$

Denote $\frac{dc_2}{2rm+dK_2} = a_1$, and let $z_2(t) = a_1 y(t)$, then (11) become

$$D^\alpha z_2(t) = z_2(t)(a_2 - z_2(t)). \tag{12}$$

Based on the methods in (Li *et al.* 2019), we also can get $\limsup_{t\to\infty} y(t) \leq \hat{m}$. Hence, the proof is completed and the region is $\Omega' = \{(S, I, y) \in R^3_+ : S(t) \leq m, I(t) \leq \frac{2rm}{d}, y(t) \leq \hat{m}\}$, where $d = \min\{r, c\}$. □

Equilibrium points

Set

$$D^\alpha S(t) = 0, D^\alpha I(t) = 0, D^\alpha y(t) = 0,$$

then the equilibrium points can be determined.
(1)The trivial equilibrium point is $E_0(0,0,0)$.
(2)The infection-free and predator-free equilibrium point is $E_1(S_1, 0, 0)$, where $S_1 = K$.
(3)The predator-only equilibrium point is $E_2(0, 0, y_2)$, where $y_2 = \frac{a_2 K_2}{c_2}$.
(4)The predator-free equilibrium point is $E_3(S_3, I_3, 0)$, where $S_3 = \frac{c}{\beta}, I_3 = \frac{r(\beta K - c)}{\beta(r+\beta K)}$. E_3 exists if $\beta > \beta_1$, where $\beta_1 = \frac{c}{K}$.
(5)The infection-free equilibrium point is $E_4(S_4, 0, y_4)$, where $S_4 = K, y_4 = \frac{a_2 K_2}{c_2}$.
(6)The interior equilibrium point is $E'(S', I', y')$, where $S' = \frac{1}{\beta}[c + \frac{c_1 a_2}{c_2}\frac{K_2+I'}{K_1+I'}], y' = \frac{a_2(I'+K_2)}{c_2}, I' = \frac{-\Delta_2 + \sqrt{\Delta_2^2 - 4\Delta_1\Delta_3}}{2\Delta_1}$, Δ_1, Δ_2 and Δ_3 are the coefficients of the equation $\Delta_1 I'^2 + \Delta_2 I' + \Delta_3 = 0$, and $\Delta_1 = \frac{r+\beta K}{K} > 0, \Delta_2 = \frac{rc_1 a_2}{K\beta c_2} + \frac{K_1(r+\beta K)}{K} + \frac{r(c-\beta K)}{\beta K}, \Delta_3 = \frac{r}{\beta K}[\frac{c_1 a_2 K_2}{c_2} + (c - \beta K)K_1]$. E' exists if $\beta > \beta_2$, where $\beta_2 = \beta_1 + \frac{c_1 a_2 K_2}{c_2 K K_1}, \beta_1 = \frac{c}{K}$.

Suppose $E^*(S^*, I^*, y^*)$ is arbitrary equilibrium point, we transform E^* into the origin. Let

$$U_1(t) = S(t) - S^*, U_2(t) = I(t) - I^*, U_3(t) = y(t) - y^*,$$

then we can rewrite system (3) as

$$\begin{aligned} D^\alpha U_1(t) &= r(U_1(t) + S^*)(1 - \frac{U_1(t) + S^* + U_2(t) + I^*}{K}) \\ &\quad - \beta(U_1(t) + S^*)(U_2(t) + I^*), \\ D^\alpha U_2(t) &= \beta(U_1(t-\tau) + S^*)(U_2(t-\tau) + I^*) \\ &\quad - c(U_2(t) + I^*) - \frac{c_1(U_2(t) + I^*)(U_3(t) + y^*)}{U_2(t) + I^* + K_1}, \\ D^\alpha U_3(t) &= (U_3(t) + y^*)(a_2 - \frac{c_2(U_3(t) + y^*)}{U_2(t) + I^* + K_2}). \end{aligned} \tag{13}$$

Taking advantage of Taylor expansion formula and linearizing the system (13), we can get

$$\begin{aligned} D^\alpha U_1(t) &= (r - \frac{2rS^*}{K} - \frac{rI^*}{K} - \beta I^*)U_1(t) \\ &\quad - (\frac{r}{K} + \beta)S^* U_2(t), \\ D^\alpha U_2(t) &= -(c + \frac{c_1 K_1 y^*}{(I^* + K_1)^2})U_2(t) - \frac{c_1 I^*}{I^* + K_1}U_3(t) \\ &\quad + \beta I^* U_1(t-\tau) + \beta S^* U_2(t-\tau), \\ D^\alpha U_3(t) &= \frac{c_2(y^*)^2}{(I^* + K_2)^2}U_2(t) + (a_2 - \frac{2c_2 y^*}{I^* + K_2})U_3(t). \end{aligned} \tag{14}$$

Stability

According to Lemma 1, we obtain

$$V_1 = \begin{pmatrix} m_{11} & m_{12} & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{32} & m_{33} \end{pmatrix}, V_2 = \begin{pmatrix} 0 & 0 & 0 \\ n_{21} & n_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{15}$$

where

$$\begin{aligned} m_{11} &= r - \frac{2rS^*}{K} - \frac{rI^*}{K} - \beta I^*, m_{12} = -(\frac{r}{K}+\beta)S^*, \\ m_{22} &= -(c + \frac{c_1 K_1 y^*}{(I^*+K_1)^2}), m_{23} = -\frac{c_1 I^*}{I^*+K_1}, \\ m_{32} &= \frac{c_2 (y^*)^2}{(I^*+K_2)^2}, m_{33} = a_2 - \frac{2c_2 y^*}{I^*+K_2}, \\ n_{21} &= \beta I^*, n_{22} = \beta S^*. \end{aligned} \tag{16}$$

Denote $V = V_1 + V_2 e^{-s\tau}$, then the Jacobi Matrix of the system (14) is

$$V = \begin{pmatrix} m_{11} & m_{12} & 0 \\ n_{21}e^{-s\tau} & m_{22} + n_{22}e^{-s\tau} & m_{23} \\ 0 & m_{32} & m_{33} \end{pmatrix}, \tag{17}$$

thus the characteristic equation of (14) can be obtained as:

$$det \begin{pmatrix} s^\alpha - m_{11} & -m_{12} & 0 \\ -n_{21}e^{-s\tau} & s^\alpha - m_{22} - n_{22}e^{-s\tau} & -m_{23} \\ 0 & -m_{32} & s^\alpha - m_{33} \end{pmatrix} = 0, \tag{18}$$

i.e.$(s^\alpha - m_{11})(s^\alpha - m_{22} - n_{22}e^{-s\tau})(s^\alpha - m_{33}) - m_{12}n_{21}e^{-s\tau}(s^\alpha - m_{33}) - m_{23}m_{32}(s^\alpha - m_{11}) = 0$.

(i) For equilibrium point $E_0(0,0,0)$, (18) becomes

$$(s^\alpha - r)(s^\alpha + c)(s^\alpha - a_2) = 0. \tag{19}$$

Suppose $s^\alpha = \lambda$, then (19) has eigenvalues $\lambda_1 = r > 0, \lambda_2 = -c < 0, \lambda_3 = a_2 > 0$, thus $|arg(\lambda_i)| = 0 < \frac{\pi\alpha}{2}, i = 1,3$. According to Lemma 1, equilibrium point E_0 is unstable.

(ii) For equilibrium point $E_1(S_1,0,0)$, (18) becomes

$$(s^\alpha + r)(s^\alpha + c - \beta K e^{-s\tau})(s^\alpha - a_2) = 0. \tag{20}$$

Let $s^\alpha = \lambda$, then (20) has a positive eigenvalue $\lambda_1 = a_2 > 0$, thus $|arg(\lambda_1)| = 0 < \frac{\pi\alpha}{2}$. According to Lemma 1, equilibrium point E_1 is unstable.

(iii) For equilibrium point $E_2(0,0,y_2)$, (18) reduces to

$$(s^\alpha - r)(s^\alpha + (c + \frac{c_1 a_2 K_2}{c_2 K_1}))(s^\alpha + a_2) = 0. \tag{21}$$

Let $s^\alpha = \lambda$, then (21) has a positive eigenvalue $\lambda_1 = r > 0$, thus $|arg(\lambda_1)| = 0 < \frac{\pi\alpha}{2}$. According to Lemma 1, equilibrium point E_2 is unstable.

(iv) For equilibrium point $E_3(S_3, I_3, 0)$, (18) reduces to

$$(s^\alpha - a_2)[(s^\alpha - m_{11})(s^\alpha - m_{22} - n_{22}e^{-s\tau}) - m_{12}n_{21}e^{-s\tau}] = 0, \tag{22}$$

where $m_{11}|_{E_3} = r - \frac{2rS_3}{K} - \frac{rI_3}{K} - \beta I_3, m_{12}|_{E_3} = -(\frac{r}{K} + \beta)S_3, m_{22}|_{E_3} = -c, m_{23}|_{E_3} = -\frac{c_1 I_3}{I_3 + K_1}, m_{32}|_{E_3} = 0, m_{33}|_{E_3} = a_2, n_{21}|_{E_3} = \beta I_3, n_{22}|_{E_3} = \beta S_3$, and $S_3 = \frac{c}{\beta}, I_3 = \frac{r(\beta K - c)}{\beta(r + \beta K)}$. Let $s^\alpha = \lambda$, then (22) has a positive eigenvalue $\lambda_1 = a_2 > 0$, thus $|arg(\lambda_1)| = 0 < \frac{\pi\alpha}{2}$. According to Lemma 1, equilibrium point E_3 is unstable.

We derive the following theorem based on the above analysis.

Theorem 9. E_0, E_1, E_2, E_3 are unstable for all $\tau \geq 0$.

(v) For equilibrium point $E_4(S_4, 0, y_4)$, (18) reduces to

$$(s^\alpha + r)(s^\alpha - m_{22} - n_{22}e^{-s\tau})(s^\alpha + a_2) = 0. \tag{23}$$

where $m_{11}|_{E_4} = -r, m_{12}|_{E_4} = -(\frac{r}{K} + \beta)K, m_{22}|_{E_4} = -(c + \frac{c_1 a_2 K_2}{c_2 K_1}), m_{23}|_{E_4} = 0, m_{32}|_{E_4} = \frac{a_2^2}{c_2}, m_{33}|_{E_4} = -a_2, n_{21}|_{E_4} = 0, n_{22}|_{E_4} = \beta K$. Let $s^\alpha = \lambda$, then two eigenvalues of (23) are $\lambda_1 = -r < 0, \lambda_2 = -a_2 < 0$, thus $|arg(\lambda_i)| = \pi > \frac{\alpha\pi}{2}, i = 1,2$. By solving the following equation

$$s^\alpha - m_{22} - n_{22}e^{-s\tau} = 0, \tag{24}$$

we can gain other eigenvalues.

When $\tau = 0$, the other eigenvalue is $\lambda_3 = (\beta K - c) - \frac{c_1 a_2 K_2}{c_2 K_1}$. $\lambda_3 < 0$ if $\beta < \beta_2 = \frac{c}{K} + \frac{c_1 a_2 K_2}{c_2 K K_1}$. Then we acquire $|arg(\lambda_i)| > \frac{\alpha\pi}{2}, i = 1,2,3$, thus all characteristic roots of (23) have negative real parts. E_4 is locally asymptotically stable on the basic of Lemma 1.

When $\tau > 0$, assume that $s = \mathrm{i}\omega = \omega(\cos\frac{\pi}{2} + \mathrm{i}\sin\frac{\pi}{2})(\omega > 0)$ is a root of (24). Separating real and imaginary parts

$$|\omega|^\alpha \cos\frac{\pi}{2}\alpha - n_{22}\cos\omega\tau - m_{22} = 0, |\omega|^\alpha \sin\frac{\pi}{2}\alpha + n_{22}\sin\omega\tau = 0. \tag{25}$$

From (25) we can obtain

$$\cos\omega\tau = \frac{1}{n_{22}}|\omega|^\alpha \cos\frac{\pi}{2}\alpha - \frac{m_{22}}{n_{22}}, \sin\omega\tau = -\frac{1}{n_{22}}|\omega|^\alpha \sin\frac{\pi}{2}\alpha. \tag{26}$$

Add up the squares of both equations of (26)

$$|\omega|^{2\alpha} - 2m_{22}\cos(\frac{\pi}{2}\alpha)|\omega|^\alpha + m_{22}^2 - n_{22}^2 = 0, \tag{27}$$

Let $\omega^\alpha = t$, then we can get

$$t^2 - 2m_{22}\cos(\frac{\pi}{2}\alpha)t + m_{22}^2 - n_{22}^2 = 0. \tag{28}$$

Since $\alpha \in (0,1], m_{22}|_{E_4} = -(c + \frac{c_1 a_2 K_2}{c_2 K_1}) < 0, n_{22}|_{E_4} = \beta K$, then $-2m_{22}\cos\frac{\pi}{2}\alpha > 0, m_{22}^2 - n_{22}^2 = (c + \frac{c_1 a_2 K_2}{c_2 K_1})^2 - \beta^2 K^2 = (K(\frac{c}{K} + \frac{c_1 a_2 K_1}{c_2 K_1 K}))^2 - \beta^2 K^2 = K^2(\beta_2)^2 - K^2\beta^2 = K^2(\beta + \beta_2)(\beta_2 - \beta)$. We derive $m_{22}^2 - n_{22}^2 > 0$ if $\beta < \beta_2$. According to Routh-Hurwitz theorem, (28) has no positive real part. Then (24) has no pure imaginary root. Therefore, equilibrium point E_4 is locally asymptotically stable. We derive the following theorem based on the above analysis.

Theorem 10. E_4 is locally asymptotically stable for $\tau \geq 0$ if $\beta < \beta_2 = \frac{c}{K} + \frac{c_1 a_2 K_2}{c_2 K K_1}$.

Furthermore, we obtain the globally asymptotically stable of system (3) at E_4. To investigate the globally asymptotically stable of system (3) at E_4, we introduce the following assumption.
(H1) $(\frac{r}{K} + \beta)S_4 - c \leq 0$,
(H2) $(c_2 y_4 - K_2 c_1)I + K_1 c_2 y_4 - K_2^2 c_1 \leq 0$.
Motivated by (Sene 2021), we define a Lyapunov functional as

$$V(t) = S(t) - S_4 - S_4 \ln\frac{S(t)}{S_4} + I(t) + y(t) - y_4 - y_4 \ln\frac{y(t)}{y_4}.$$

Taking fractional-order derivative on both sides, according to Lemma 4, we get

$$
\begin{aligned}
D^{\alpha}V(t) \leq &(\frac{S(t)-S_4}{S(t)})D^{\alpha}S(t) + D^{\alpha}I(t) + \frac{y(t)-y_4}{y(t)}D^{\alpha}y(t) \\
&= (S(t)-S_4)(r(1-\frac{S(t)+I(t)}{K}) - \beta I(t)) + \\
&(\beta S(t-\tau)I(t-\tau) - cI(t) - \frac{c_1 I(t)y(t)}{I(t)+K_1}) + \\
&(y(t)-y_4)(a_2 - \frac{c_2 y(t)}{I(t)+K_2}) \\
&= (S(t)-S_4)(-\frac{r}{K}(S(t)-S_4) - (\frac{r}{K}+\beta)I(t)) \\
&+ (\beta S(t-\tau)I(t-\tau) - cI(t) - \frac{c_1 I(t)y(t)}{I(t)+K_1}) \\
&+ (y(t)-y_4)(\frac{c_2 y_4}{K_2} - \frac{c_2 y(t)}{I(t)+K_2}) \\
&= -\frac{r}{K}(S(t)-S_4)^2 - (\frac{r}{K}+\beta)I(t)(S(t)-S_4) \\
&+ \beta S(t-\tau)I(t-\tau) - cI(t) - \frac{c_1 I(t)y(t)}{I(t)+K_1} + \\
&c_2(y(t)-y_4)(\frac{y_4}{K_2} - \frac{y(t)}{I(t)+K_2}) \\
&= -\frac{r}{K}(S(t)-S_4)^2 - (\frac{r}{K}+\beta)S(t)I(t) + \\
&(\frac{r}{K}+\beta)S_4 I(t) + \beta S(t-\tau)I(t-\tau) \\
&- cI(t) - \frac{c_1 I(t)y(t)}{I(t)+K_1} + c_2(y(t)-y_4)\times \\
&(-\frac{y(t)-y_4}{I(t)+K_2} + \frac{I(t)y_4}{K_2(I(t)+K_2)}) \\
&= -\frac{r}{K}(S(t)-S_4)^2 + (-(\frac{r}{K}+\beta)S(t)I(t) + \\
&\beta S(t-\tau)I(t-\tau)) + ((\frac{r}{K}+\beta)S_4 - c)I(t) \\
&- \frac{c_2}{K_2+I(t)}(y(t)-y_4)^2 - \frac{c_2 I(t)(y_4)^2}{K_2(I(t)+K_2)} \\
&+ \frac{c_2 y_4 I(t)y(t)}{K_2(I(t)+K_2)} - \frac{c_1 I(t)y(t)}{I(t)+K_1} \\
&= -\frac{r}{K}(S(t)-S_4)^2 + (\beta - (\frac{r}{K}+\beta))S(t)I(t) \\
&+ ((\frac{r}{K}+\beta)S_4 - c)I(t) - \frac{c_2}{K_2+I(t)}\times \\
&(y(t)-y_4)^2 - \frac{c_2 I(t)(y_4)^2}{K_2(I(t)+K_2)} + \\
&\frac{(c_2y_4 - K_2c_1)I(t) + (K_1c_2y_4 - K_2^2c_1)}{K_2(I(t)+K_2)(I(t)+K_1)}I(t)y(t) \\
&= -\frac{r}{K}(S(t)-S_4)^2 - \frac{r}{K}S(t)I(t) \\
&+ ((\frac{r}{K}+\beta)S_4 - c)I(t) - \frac{c_2}{K_2+I(t)}(y(t)-y_4)^2 \\
&- \frac{c_2 I(t)(y_4)^2}{K_2(I(t)+K_2)} + \\
&\frac{(c_2y_4 - K_2c_1)I(t) + (K_1c_2y_4 - K_2^2c_1)}{K_2(I(t)+K_2)(I(t)+K_1)}I(t)y(t).
\end{aligned}
$$

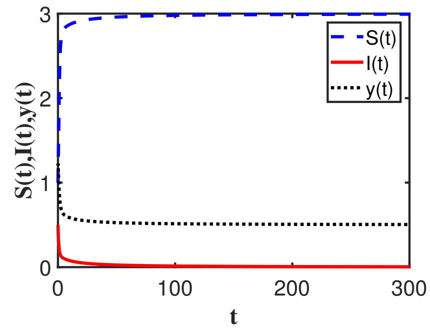


Figure 1 Waveform plots of system (49) with $\tau = 0.4$.

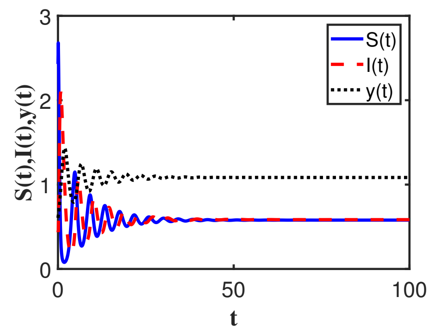


Figure 2 Waveform plots of system (50) with $\tau = 0 < \tau_0$.

Based on the assumption $(\frac{r}{K}+\beta)S_4 - c \leq 0$ and $(c_2y_4 - K_2c_1)I + K_1c_2y_4 - K_2^2c_1 \leq 0$, we can get $D^{\alpha}V(t) \leq 0$. According to (Huo *et al.* 2015), we can derive the system (3) is globally asymptotically stable at E_4.

Therefore, We derive the following theorem.

Theorem 11. Assume that $(\frac{r}{K}+\beta)S_4 - c \leq 0$ and $(c_2y_4 - K_2c_1)I + K_1c_2y_4 - K_2^2c_1 \leq 0$, then the system (3) is globally asymptotically stable at E_4.

(vi) For equilibrium point $E'(S', I', y')$, the characteristic equation at E' is:

$$s^{3\alpha} + \delta_2 s^{2\alpha} + \delta_1 s^{\alpha} + \delta_0 + e^{-s\tau}(\vartheta_2 s^{2\alpha} + \vartheta_1 s^{\alpha} + \vartheta_0) = 0, \tag{29}$$

where

$$
\begin{aligned}
\delta_2 &= -(m_{11} + m_{22} + m_{33}), \\
\delta_1 &= m_{11}m_{22} + m_{22}m_{33} + m_{11}m_{33} - m_{23}m_{32}, \\
\delta_0 &= m_{11}m_{23}m_{32} - m_{11}m_{22}m_{33},
\end{aligned}
$$

$$
\begin{aligned}
\vartheta_2 &= -n_{22}, \\
\vartheta_1 &= m_{11}n_{22} - m_{12}n_{21} + m_{33}n_{22}, \\
\vartheta_0 &= m_{12}m_{33}n_{21} - m_{11}m_{33}n_{22}.
\end{aligned}
$$

When $\tau = 0$, (29) can be expressed as

$$s^{3\alpha} + (\delta_2 + \vartheta_2)s^{2\alpha} + (\delta_1 + \vartheta_1)s^{\alpha} + \delta_0 + \vartheta_0 = 0, \tag{30}$$

Let $z = s^{\alpha}$, then

$$z^3 + (\delta_2 + \vartheta_2)z^2 + (\delta_1 + \vartheta_1)z + \delta_0 + \vartheta_0 = 0. \tag{31}$$

According to the Routh-Hurwitz theorem, (30) has no positive real part if $\delta_2 + \vartheta_2 > 0$ and $(\delta_2 + \vartheta_2)(\delta_1 + \vartheta_1) - \delta_0 + \vartheta_0 > 0$. Thus (29) has no pure imaginary root. Hence, E' is locally asymptotically stable.

We obtain the following theorem on the basic of our analysis.

Theorem 12. The equilibrium point E' is locally asymptotically stable for $\tau = 0$ if $\delta_2 + \vartheta_2 > 0$ and $(\delta_2 + \vartheta_2)(\delta_1 + \vartheta_1) - \delta_0 + \vartheta_0 > 0$.

Assume that $s = i\xi = \xi(\cos\frac{\pi}{2} + i\sin\frac{\pi}{2})(\xi > 0)$ is a root of (29). Separating real and imaginary parts,

$$\Psi\cos\xi\tau + \Omega\sin\xi\tau = \Phi_1, \tag{32}$$

$$\Omega\cos\xi\tau - \Psi\sin\xi\tau = \Phi_2, \tag{33}$$

where

$$\begin{aligned}
\Psi &= \vartheta_0 + \vartheta_1\xi^{\alpha}\cos(\alpha\frac{\pi}{2}) + \vartheta_2\xi^{2\alpha}\cos(2\alpha\frac{\pi}{2}),\\
\Omega &= \vartheta_1\xi^{\alpha}\sin(\alpha\frac{\pi}{2}) + \vartheta_2\xi^{2\alpha}\sin(2\alpha\frac{\pi}{2}),\\
\Phi_1 &= -(\delta_0 + \delta_1\xi^{\alpha}\cos(\alpha\frac{\pi}{2}) + \delta_2\xi^{2\alpha}\cos(2\alpha\frac{\pi}{2}) +\\
&\quad \xi^{3\alpha}\cos(3\alpha\frac{\pi}{2})),\\
\Phi_2 &= -(\delta_1\xi^{\alpha}\sin(\alpha\frac{\pi}{2}) + \delta_2\xi^{2\alpha}\sin(2\alpha\frac{\pi}{2}) +\\
&\quad \xi^{3\alpha}\sin(3\alpha\frac{\pi}{2})).
\end{aligned}$$

Add up the squares of both equations (32) and (33),

$$\begin{aligned}
G(\xi^{\alpha}) &= \xi^{6\alpha} + H_5\xi^{5\alpha} + H_4\xi^{4\alpha} + H_3\xi^{3\alpha}\\
&\quad + H_2\xi^{2\alpha} + H_1\xi^{\alpha} + H_0\\
&= 0,
\end{aligned} \tag{34}$$

where

$$\begin{aligned}
H_5 &= 2\delta_2\cos(\alpha\frac{\pi}{2}),\\
H_4 &= \delta_2^2 - \vartheta_2^2 + 2\delta_1\cos(2\alpha\frac{\pi}{2}),\\
H_3 &= (2\delta_1\delta_2 - 2\vartheta_1\vartheta_2)\cos(\alpha\frac{\pi}{2}) + 2\delta_0\cos(3\alpha\frac{\pi}{2}),\\
H_2 &= \delta_1^2 - \vartheta_1^2 + (2\delta_0\delta_2 - 2\vartheta_0\vartheta_2)\cos(2\alpha\frac{\pi}{2}),\\
H_1 &= (2\delta_0\delta_1 - 2\vartheta_0\vartheta_1)\cos(\alpha\frac{\pi}{2}),\\
H_0 &= \delta_0^2 - \vartheta_0^2.
\end{aligned}$$

According to the Routh-Hurwitz theorem, we can get the routh list

$$\begin{array}{cccc}
1 & H_4 & H_2 & H_0\\
H_5 & H_3 & H_1 & 0\\
b_5 & b_3 & b_1 & 0\\
d_5 & d_3 & 0 & 0\\
u_5 & u_3 & 0 & 0\\
v_5 & 0 & 0 & 0\\
h_5 & & &
\end{array} \tag{35}$$

where $b_5 = -\frac{H_3 - H_4H_5}{H_5}, b_3 = -\frac{H_1 - H_2H_5}{H_5}, b_1 = H_0, d_5 = -\frac{H_5b_3 - H_3b_5}{b_5}, d_3 = -\frac{H_5b_1 - H_1b_5}{b_5}, u_5 = -\frac{b_5d_3 - b_3d_5}{d_5}, u_3 = b_1, v_5 = -\frac{d_5u_3 - d_3u_5}{u_5}, h_5 = u_3$.

When (35) satisfies some conditions(Li *et al.* 2021), there will be a change of sign, then (34) at least has one positive root. Thus, there exists a pair of purely imaginary roots of (29), which satisfy one of the conditions of Hopf bifurcation.

From (32) and (33), we can derive

$$\begin{aligned}
\cos\xi\tau &= \frac{\Psi\Phi_1 + \Omega\Phi_2}{\Omega^2 + \Psi^2},\\
\sin\xi\tau &= \frac{\Omega\Phi_1 - \Psi\Phi_2}{\Omega^2 + \Psi^2}.
\end{aligned} \tag{36}$$

According to (36), we can get

$$\tau^{(k)} = \frac{1}{\xi}(\arctan\frac{\Omega\Phi_1 - \Psi\Phi_2}{\Psi\Phi_1 + \Omega\Phi_2} + k\pi), k = 0, 1, 2, \ldots, \tag{37}$$

then we define the bifurcation point

$$\tau_0 = \min\tau^{(k)}, k = 0, 1, 2, \ldots \tag{38}$$

We introduce the following assumption to obtain the conditions of Hopf bifurcation.

(H3) $\frac{A_1N_1 + A_2N_2}{N_1^2 + N_2^2} \neq 0$,

where A_1, A_2 are defined by (43), and N_1, N_2 are defined by (48).

Lemma 13. Let $s(\tau) = \gamma(\tau) + i\omega(\tau)$ be the root of (29) near $\tau = \tau_j$ meeting $\gamma(\tau_j) = 0$ and $\omega(\tau_j) = \omega_0$, then the following transversality condition meets

$$Re[\frac{ds}{d\tau}]|_{\tau=\tau_0, \omega=\omega_0} \neq 0. \tag{39}$$

Proof. Let $P_1(s) = s^{3\alpha} + \delta_2s^{2\alpha} + \delta_1s^{\alpha} + \delta_0, P_2(s) = \vartheta_2s^{2\alpha} + \vartheta_1s^{\alpha} + \vartheta_0$, then (29) can be rewritten as

$$P_1(s) + P_2(s)e^{-s\tau} = 0. \tag{40}$$

Derivation on both sides of (40) respect to τ,

$$P_1'(s)\frac{ds}{d\tau} + P_2'(s)e^{-s\tau}\frac{ds}{d\tau} + P_2(s)e^{-s\tau}(-\tau\frac{ds}{d\tau} - s) = 0, \tag{41}$$

where $P_i'(s)$ are the derivatives of $P_i(s)(i = 1, 2)$.

Then,

$$\frac{ds}{d\tau} = \frac{M(s)}{N(s)}, \tag{42}$$

where

$$\begin{aligned}
M(s) &= s(\vartheta_2s^{2\alpha} + \vartheta_1s^{\alpha} + \vartheta_0)e^{-s\tau},\\
N(s) &= 3\alpha s^{3\alpha-1} + 2\alpha\delta_2s^{2\alpha-1} + \alpha\delta_1s^{\alpha-1}\\
&\quad - \tau e^{-s\tau}(\vartheta_2s^{2\alpha} + \vartheta_1s^{\alpha} + \vartheta_0)\\
&\quad + e^{-s\tau}(2\alpha\vartheta_2s^{2\alpha-1} + \alpha\vartheta_1s^{\alpha-1}).
\end{aligned}$$

By straightforward computation,

$$[\frac{ds}{d\tau}]|_{\tau=\tau_0, \omega=\omega_0} = \frac{A_1 + iA_2}{(B_1 + C_1 + D_1) + i(B_2 + C_2 + D_2)},$$

where

$$\begin{aligned}
A_1 &= (-\vartheta_2\omega_0^{2\alpha+1}\sin(\frac{\pi}{2}2\alpha) - \vartheta_1\omega_0^{\alpha+1}sin(\frac{\pi}{2}\alpha))\times\\
&\quad \cos(\omega_0\tau_0) + (\vartheta_2\omega_0^{2\alpha+1}\cos(\frac{\pi}{2}2\alpha) +\\
&\quad \vartheta_1\omega_0^{\alpha+1}\cos(\frac{\pi}{2}\alpha) + \omega_0\vartheta_0)\sin(\omega_0\tau_0),\\
A_2 &= (\vartheta_2\omega_0^{2\alpha+1}\sin(\frac{\pi}{2}2\alpha) + \vartheta_1\omega_0^{\alpha+1}\sin(\frac{\pi}{2}\alpha))\times\\
&\quad \sin(\omega_0\tau_0) + (\vartheta_2\omega_0^{2\alpha+1}\cos(\frac{\pi}{2}2\alpha) +\\
&\quad \vartheta_1\omega_0^{\alpha+1}\cos(\frac{\pi}{2}\alpha) + \omega_0\vartheta_0)\cos(\omega_0\tau_0),
\end{aligned} \tag{43}$$

$$\begin{aligned}
B_1 &= 3\alpha\omega_0^{3\alpha-1}\cos(\frac{(3\alpha-1)\pi}{2}) + 2\alpha\delta_2\omega_0^{2\alpha-1}\times \\
&\quad \cos(\frac{(2\alpha-1)\pi}{2}) + \alpha\delta_1\omega_0^{\alpha-1}\cos(\frac{(\alpha-1)\pi}{2}), \\
B_2 &= 3\alpha\omega_0^{3\alpha-1}sin(\frac{(3\alpha-1)\pi}{2}) + 2\alpha\delta_2\omega_0^{2\alpha-1}\times \\
&\quad \sin(\frac{(2\alpha-1)\pi}{2}) + \alpha\delta_1\omega_0^{\alpha-1}\sin(\frac{(\alpha-1)\pi}{2}),
\end{aligned} \tag{44}$$

$$\begin{aligned}
C_1 &= -\tau sin\omega_0\tau_0(\vartheta_2\omega_0^{2\alpha}\sin(\frac{\pi}{2}2\alpha) + \vartheta_1\omega_0^{\alpha}\sin(\frac{\pi}{2}\alpha)) \\
&\quad - \tau\cos\omega_0\tau_0(\vartheta_2\omega_0^{2\alpha}\cos(\frac{\pi}{2}2\alpha) + \vartheta_1\omega_0^{\alpha}\cos(\frac{\pi}{2}\alpha) \\
&\quad + \vartheta_0), \\
C_2 &= -\tau\cos\omega_0\tau_0(\vartheta_2\omega_0^{2\alpha}\sin(\frac{\pi}{2}2\alpha) + \vartheta_1\omega_0^{\alpha}\sin(\frac{\pi}{2}\alpha)) \\
&\quad + \tau sin\omega_0\tau_0(\vartheta_2\omega_0^{2\alpha}\cos(\frac{\pi}{2}2\alpha) + \vartheta_1\omega_0^{\alpha}\cos(\frac{\pi}{2}\alpha) \\
&\quad + \vartheta_0),
\end{aligned} \tag{45}$$

$$\begin{aligned}
D_1 &= \cos(\omega_0\tau_0)(2\alpha\vartheta_2\omega_0^{2\alpha-1}\times \\
&\quad \cos\frac{(2\alpha-1)\pi}{2} + \alpha\vartheta_1\omega_0^{\alpha-1}\cos\frac{(\alpha-1)\pi}{2})+ \\
&\quad \sin(\omega_0\tau_0)(2\alpha\vartheta_2\omega_0^{2\alpha-1}\sin\frac{(2\alpha-1)\pi}{2}+ \\
&\quad \alpha\vartheta_1\omega_0^{\alpha-1}\sin\frac{(\alpha-1)\pi}{2}), \\
D_2 &= -\sin(\omega_0\tau_0)(2\alpha\vartheta_2\omega_0^{2\alpha-1}\cos\frac{(2\alpha-1)\pi}{2}+ \\
&\quad \alpha\vartheta_1\omega_0^{\alpha-1}\cos\frac{(\alpha-1)\pi}{2}) + \cos(\omega_0\tau_0)\times \\
&\quad (2\alpha\vartheta_2\omega_0^{2\alpha-1}\sin\frac{(2\alpha-1)\pi}{2}+ \\
&\quad \alpha\vartheta_1\omega_0^{\alpha-1}\sin\frac{(\alpha-1)\pi}{2}).
\end{aligned} \tag{46}$$

Hence,

$$Re[\frac{ds}{d\tau}]|_{\tau=\tau_0,\omega=\omega_0} = \frac{A_1N_1 + A_2N_2}{N_1^2 + N_2^2}, \tag{47}$$

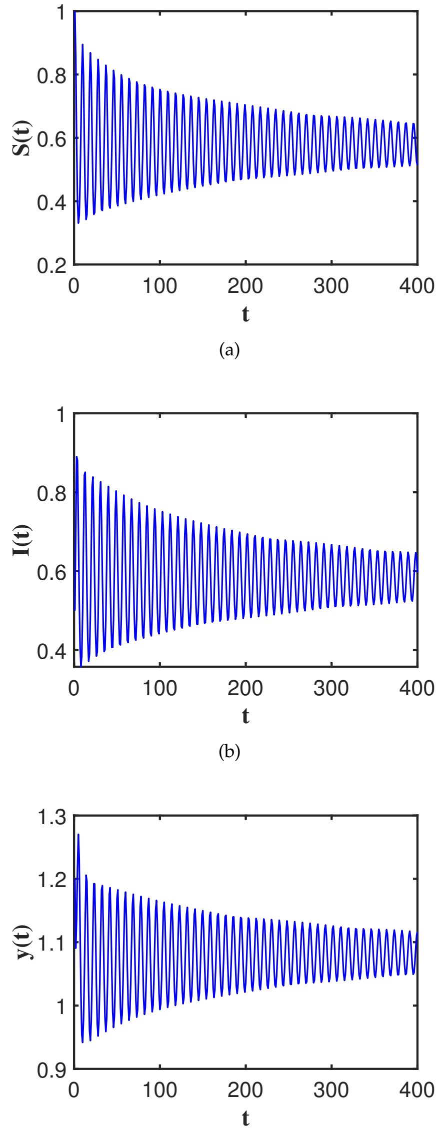
where

$$N_1 = B_1 + C_1 + D_1, N_2 = B_2 + C_2 + D_2. \tag{48}$$

The proof is completed. □

Hence, we obtain the following theorem.

Theorem 14. Suppose that (H3) holds, we can gain the following results:
(i) E' is locally asymptotically stable for $\tau \in [0, \tau_0)$.
(ii) System (3) undergoes a Hopf bifurcation at E' when $\tau = \tau_0$.



(a)

(b)

(c)

Figure 3 Waveform plots of system (50) with $\tau = 0.1 < \tau_0$.

NUMERICAL SIMULATIONS

Diethem et al proposed the Adams-Bashforth-Moulton prediction-correction numerical algorithm of fractional differential equations defined by Caputo(Kai *et al.* 2002), and Bhalekar et al extended it to fractional differential equations with delay(Bhalekar and Daftardar-Gejji 2011). Here, the modified Adams-Bashforth-Moulton prediction-correction numerical algorithm is used to verify our theoretical analysis(Bhalekar and Daftardar-Gejji 2011).

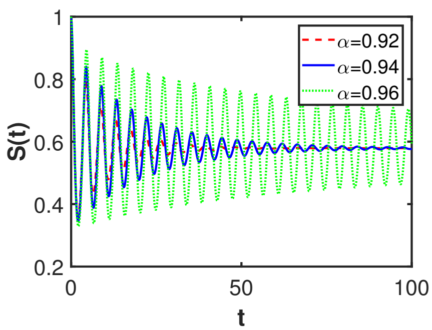
Example 1

According to the numerical simulations of (Zhou *et al.* 2010) and (Adak *et al.* 2020), we make two examples and set the following values for the parameters. When the order is close to 1, the dynamic properties of fractional-order system will be close to the dynamic properties of integer-order system. Hence, we choose the order $\alpha = 0.96$ and the other parameters are taken from (Zhou *et al.* 2010),$r = 2, a_2 = 1, c = 0.3, c_1 = 1, c_2 = 1, K = 3, K_1 = 0.6, K_2 =$

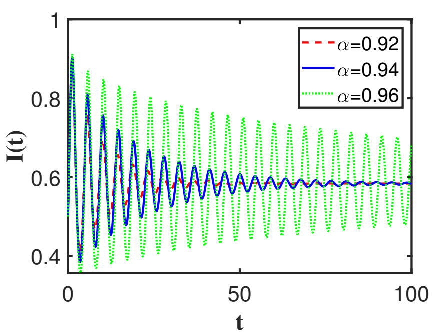
0.5. Then, we choose $\beta = 0.37 < \beta_2$, which satisfy the Theorem 10, then system (3) is

$$
\begin{aligned}
D^{0.96}S(t) &= 2S(t)(1 - \frac{S(t)+I(t)}{3}) - 0.37S(t)I(t), \\
D^{0.96}I(t) &= 0.37S(t-\tau)I(t-\tau) - 0.3I(t) - \frac{I(t)y(t)}{I(t)+0.6}, \\
D^{0.96}y(t) &= y(t)(1 - \frac{y(t)}{I(t)+0.5}).
\end{aligned}
\tag{49}
$$

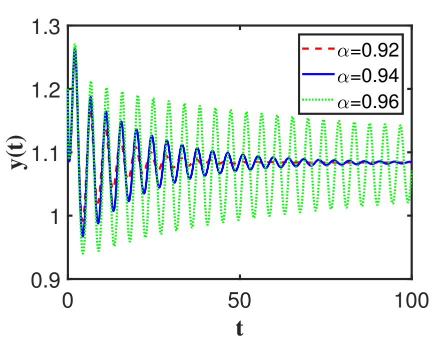
It is not difficult to get equilibrium point $E_4(S_4, I_4, y_4) = (3, 0, 0.5)$. Fig. 1 exhibits that E_4 is locally asymptotically stable.



(a)



(b)



(c)

Figure 4 Waveform plots of system (50) with $\tau = 0.1$ for $\alpha = 0.92, \alpha = 0.94, \alpha = 0.96$.

Example 2

Choose $\beta = 2.1 > \beta_2$, thus E' exists. The system (3) is

$$
\begin{aligned}
D^{0.96}S(t) &= 2S(t)(1 - \frac{S(t)+I(t)}{3}) - 2.1S(t)I(t), \\
D^{0.96}I(t) &= 2.1S(t-\tau)I(t-\tau) - 0.3I(t) - \frac{I(t)y(t)}{I(t)+0.6}, \\
D^{0.96}y(t) &= y(t)(1 - \frac{y(t)}{I(t)+0.5}).
\end{aligned}
\tag{50}
$$

We acquire $E'(S', I', y') = (0.5788, 0.5834, 1.0834)$. It is not difficult to check system (50) satisfys $\delta_2 + \vartheta_2 = 0.9345 > 0$ and $(\delta_2 + \vartheta_2)(\delta_1 + \vartheta_1) - (\delta_0 + \vartheta_0) = 0.0923 > 0$. Thus, system (50) at E' is locally asymptotically stable for $\tau = 0$. We calculate that $\omega_0 = 1.3490, \tau_0 = 0.1689$. Fig. 2 and Fig. 3 show that E' is locally asymptotically stable when $\tau = 0 < \tau_0$ and $\tau = 0.1 < \tau_0$. For Fig. 3, we draw waveform plots every 20 points as a point. Motivated by the investigation on the different orders in (Sene 2019) and (Sene 2022), we show that the waveform plots of system (50) with $\tau = 0.1$ for different orders α in Fig. 4. The numerical simulation results implies that the lower values of α, the oscillating behavior is suppressed. E' is unstable of system (50) when $\tau = 0.2 > \tau_0$, which is shown in Fig. 5. Here, we give the waveform plot of $S(t)$. The waveform plots of $I(t)$ and $y(t)$ are omitted. Furthermore, we give the phase portraits in I-y plane for $\tau = 1, \tau = 3$ and $\tau = 6$. Fig. 6 exhibits the development of chaos.

Remark. In system (3), the order is $0 < \alpha \leq 1$. When $\alpha = 1$, this system is reduced to system (2). Therefore, our research extends the results of system (2).

Remark. The difference between the integer-order system (2) and the fractional-order system (3) are as follows. E_0, E_1, E_2, E_3 of system (3) are unstable for all $\tau \geq 0$, and if $\beta \leq \beta_2$, equilibrium point E_4 is locally asymptotically stable for $\tau \geq 0$. In integer-order system (2), it also has the same results. However, the conditions of the global asymptotically stability for equilibrium point E_4 is different from system (2). And the conditions of the occurrence of Hopf bifurcation of equilibrium point E' are related to the order α, which is different from integer-order system (2). Besides, the numerical results indicate that the oscillation behavior is suppressed when the order α is lower. And the chaos gradually arise when the delay τ increases. These results are not shown in the integer-order system (2).

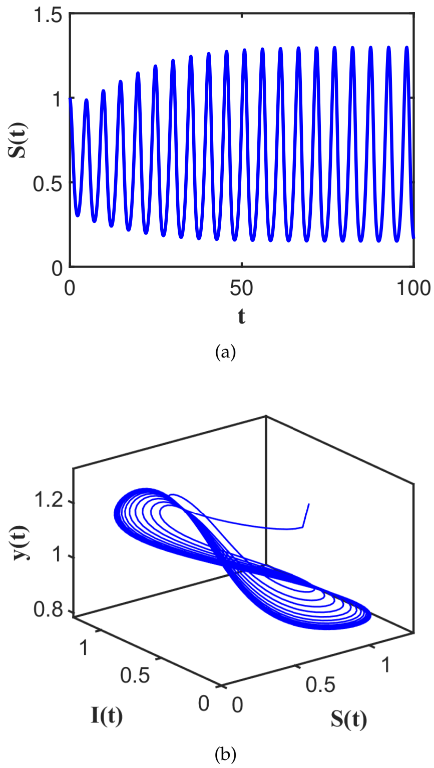
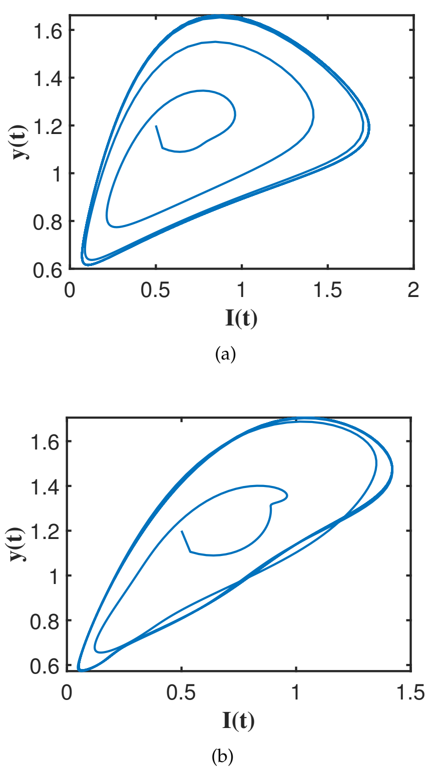


Figure 5 Waveform plots of system (50) with $\tau = 0.2 > \tau_0$.



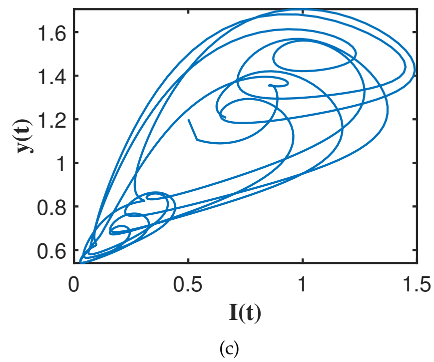


Figure 6 Phase portraits of system (50) in I-y plane for $\tau = 1, \tau = 3, \tau = 6$ respectively.

CONCLUSION

A fractional-order Leslie-Gower prey-predator-parasite system with delay is considered in this article. We investigate the existence and uniqueness of the solutions, as well as non-negativity and boundedness. We also show E_0, E_1, E_2, E_3 are unstable for $\tau \geq 0$ and if $\beta < \beta_2$, E_4 is locally asymptotically stable for $\tau \geq 0$. If the conditions of Theorem 10 are meeted, the system (3) at E_4 is globally asymptotically stable. If the conditions of Theorem 12 are satisfied, E' is locally asymptotically stable for $\tau = 0$ by Routh-Hurwitz theorem. In addition, E' occurs Hopf bifurcation when the conditions of Theorem 14 are meeted. We can change the critical value τ_0 to control the stability of system. Moreover, the system exhibits different results for different order α. The numerical results indicate that the oscillation behavior is suppressed for $\tau = 0.1$ when the order α is lower. The chaos gradually arise when the delay τ increases. Finally, we hope to explore chaos of this system.

Acknowledgments

The research is supported by National Natural Science Foundation of China (No.12172340). Liguo Yuan is a visiting scholar of Anhui University from September 2021 to August 2022.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Availability of data and material

Not applicable.

LITERATURE CITED

Adak, D., N. Bairagi, and H. Robert, 2020 Chaos in delay-induced leslie–gower prey–predator–parasite model and its control through prey harvesting. Nonlinear Analysis: Real World Applications **51**: 102998.

Alidousti, J. and M. Mostafavi Ghahfarokhi, 2019 Stability and bifurcation for time delay fractional predator prey system by incorporating the dispersal of prey. Applied Mathematical Modelling **72**: 385–402.

Anderson, R. M. and R. M. May, 1980 Infectious diseases and population cycles of forest insects. Science **210**: 658–661.

Bhalekar, S. and V. Daftardar-Gejji, 2011 A predictor-corrector scheme for solving nonlinear delay differential equations of fractional order. Fractional Calculus and Applications **5**: 1–9.

Boukhouima, A., K. Hattaf, and N. Yousfi, 2017 Dynamics of a fractional order hiv infection model with specific functional response and cure rate. International Journal of Differential Equations **2017**: 8372140.
Chinnathambi, R. and F. A. Rihan, 2018 Stability of fractional-order prey–predator system with time-delay and monod–haldane functional response. Nonlinear Dynamics **92**: 1637–1648.
Cruz, V.-D.-L., 2015 Volterra-type lyapunov functions for fractional-order epidemic systems. Communications in Nonlinear Science and Numerical Simulation **24**: 75–85.
Deng, W., C. Li, and J. Lü, 2007 Stability analysis of linear fractional differential system with multiple time delays. Nonlinear Dynamics **48**: 409–416.
Fernández-Carreón, B., J. Munoz-Pacheco, E. Zambrano-Serrano, and O. Félix-Beltrán, 2022 Analysis of a fractional-order glucose-insulin biological system with time delay. Chaos Theory and Applications **4**: 10–18.
Hu, T. C., D. L. Qian, and C. P. Li, 2009 Comparison theorems for fractional differential equations. Communication on Applied Mathematics and Computation **23**: 97–103.
Huang, C., H. Li, T. Li, and S. Chen, 2019 Stability and bifurcation control in a fractional predator–prey model via extended delay feedback. International Journal of Bifurcation and Chaos **29**: 1950150.
Huang, C., H. Liu, X. Chen, M. Zhang, L. Ding, *et al.*, 2020 Dynamic optimal control of enhancing feedback treatment for a delayed fractional order predator–prey model. Physica A: Statistical Mechanics and its Applications **554**: 124136.
Huo, J., H. Zhao, and L. Zhu, 2015 The effect of vaccines on backward bifurcation in a fractional order hiv model. Nonlinear Analysis: Real World Applications **26**: 289–305.
Kai, D., N. J. Ford, and A. D. Freed, 2002 A predictor-corrector approach for the numerical solution of fractional differential equations. Nonlinear Dynamics **29**: 3–22.
Kashkynbayev, A. and F. A. Rihan, 2021 Dynamics of fractional-order epidemic models with general nonlinear incidence rate and time-delay. Mathematics **9**.
Kilbas, A. A., H. M. Srivastava, and J. J. Trujillo, 2006 *Theory and Applications of Fractional Differential Equations*. Elsevier.
Li, C. and G. Chen, 2004 Chaos in the fractional order chen system and its control. Chaos, Solitons & Fractals **22**: 549–554.
Li, H., L. Zhang, C. Hu, Y. Jiang, and Z. Teng, 2017a Dynamical analysis of a fractional-order predator-prey model incorporating a prey refuge. Journal of Applied Mathematics and Computing **54**: 435–449.
Li, H., L. Zhang, C. Hu, Y. Jiang, and Z. Teng, 2017b Dynamical analysis of a fractional-order predator-prey model incorporating a prey refuge. Journal of Applied Mathematics and Computing **54**: 435–449.
Li, S., C. Huang, and X. Song, 2019 Bifurcation based-delay feedback control strategy for a fractional-order two-prey one-predator system. Complexity **2019**: 1–13.
Li, X. L., F. Gao, and W. Q. Li, 2021 Stability analysis of fractional-oeder hepatitis b virus infection model with immune delay(in Chinese). Acta Mathematica Scientia **41**: 562–576.
Li, Y., Y. Chen, and I. Podlubny, 2010 Stability of fractional-order nonlinear dynamic systems: Lyapunov direct method and generalized mittag–leffler stability. Computers & Mathematics with Applications **59**: 1810–1821, Fractional Differentiation and Its Applications.
Mahmoud, G. M., A. A. Arafa, T. M. Abed-Elhameed, and E. E. Mahmoud, 2017 Chaos control of integer and fractional orders of chaotic burke–shaw system using time delayed feedback control. Chaos, Solitons & Fractals **104**: 680–692.
Mbava, W., J. Mugisha, and J. Gonsalves, 2017 Prey, predator and super-predator model with disease in the super-predator. Applied Mathematics and Computation **297**: 92–114.
Moustafa, M., M. H. Mohd, A. I. Ismail, and F. A. Abdullah, 2020 Dynamical analysis of a fractional-order eco-epidemiological model with disease in prey population. Advances in Difference Equations **2020**: 1–24.
Odibat, Z. M. and N. T. Shawagfeh, 2007 Generalized taylor's formula. Applied Mathematics and Computation **186**: 286–293.
Pu, W., 2020 Stability analysis of a class of fractional sis models with time delay (in chinese). Journal of Tonghua Normal University **41**: 18–22.
Rajagopal, K., N. Hasanzadeh, and F. Parastesh, 2020 A fractional-order model for the novel coronavirus (covid-19) outbreak. Nonlinear Dynamics **101**: 711–718.
Rihan, F. and C. Rajivganthi, 2020 Dynamics of fractional-order delay differential model of prey-predator system with holling-type iii and infection among predators. Chaos, Solitons & Fractals **141**: 110365.
Sene, N., 2019 Stability analysis of the generalized fractional differential equations with and without exogenous inputs. Journal of Nonlinear Sciences and Applications **12**: 562–572.
Sene, N., 2021 Qualitative analysis of class of fractional-order chaotic system via bifurcation and lyapunov exponents notions. Journal of Mathematics **2021**.
Sene, N., 2022 Fractional model and exact solutions of convection flow of an incompressible viscous fluid under the newtonian heating and mass diffusion. Journal of Mathematics **2022**: 1–20.
Shaikh, A. A., H. Das, and N. Ali, 2018 Study of lg-holling type iii predator-prey model with disease in predator. Journal of Applied Mathematics and Computing **58**: 235–255.
Shi, J., K. He, and H. Fang, 2022 Chaos, hopf bifurcation and control of a fractional-order delay financial system. Mathematics and Computers in Simulation **194**: 348–364.
Tao, B., M. Xiao, Q. Sun, and J. Cao, 2018 Hopf bifurcation analysis of a delayed fractional-order genetic regulatory network model. Neurocomputing **275**: 677–686.
Wang, Z., M. Du, and M. Shi, 2011 Stability test of fractional delay systems via integration. Science China: Physics, Mechanics and Astronomy **54**: 1839–1846.
Xu, R. and S. Zhang, 2013 Modelling and analysis of a delayed predator–prey model with disease in the predator. Applied Mathematics and Computation **224**: 372–386.
Yousef, F. B., A. Yousef, and C. Maji, 2021 Effects of fear in a fractional-order predator-prey system with predator density-dependent prey mortality. Chaos, Solitons & Fractals **145**: 110711.
Yuan, L., Q. Yang, and C. Zeng, 2013 Chaos detection and parameter identification in fractional-order chaotic systems with delay. Nonlinear Dynamics **73**: 439–448.
Zhou, X., J. Cui, X. Shi, and X. Song, 2010 A modified leslie–gower predator–prey model with prey infection. Journal of Applied Mathematics and Computing **33**: 471–487.

How to cite this article: Yang, X., Yuan, L., and Wei, Z. Stability and Hopf Bifurcation Analysis of a Fractional-order Leslie-Gower Prey-predator-parasite System with Delay. Chaos Theory and Applications, 4(2), 71-81, 2022.

CHAOS Theory and Applications in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.4 / No.2 / 2022 / pp.82-87
https://doi.org/10.51537/chaos.1114878

CNN-Based Approach for Overlapping Erythrocyte Counting and Cell Type Classification in Peripheral Blood Images

Muhammed Ali Pala [*,β,1], **Murat Erhan Çimen** [*,2], **Mustafa Zahid Yıldız** [*,β,3], **Gökçen Çetinel** [§,4], **Emir Avcıoğlu** [α,5] and **Yusuf Alaca** [‡,6]

[*,†,‡]Department of Electric and Electronic Engineering, Sakarya University of Applied Sciences, 54187, Sakarya, Turkey, [§]Department of Electric and Electronic Engineering, Sakarya University, 54187, Sakarya, Turkey, [α]Department of Mechanical Engineering, Hitit University, 19030, Çorum, Turkey, [‡]Department of Computer Technology, Hitit University, 19169, Çorum, Turkey, [β]Biomedical Technologies Application and Research Center (BIYOTAM), Sakarya University of Applied Sciences, Sakarya, Turkey.

ABSTRACT Classification and counting of cells in the blood is crucial for diagnosing and treating diseases in the clinic. A peripheral blood smear method is a fast, reliable, robust diagnostic tool for examining blood samples. However, cell overlap during the peripheral smear process may cause incorrectly predicted results in counting blood cells and classifying cell types. The overlapping problem can occur in automated systems and manual inspections by experts. Convolutional neural networks (CNN) provide reliable results for the segmentation and classification of many problems in the medical field. However, creating ground truth labels in the data during the segmentation process is time-consuming and error-prone. This study proposes a new CNN-based strategy to eliminate the overlap-induced counting problem in peripheral smear blood samples and accurately determine the blood cell type. In the proposed method, images of the peripheral blood were divided into sub-images, block by block, using adaptive image processing techniques to identify the overlapping cells and cell types. CNN was used to classify cell types and overlapping cell numbers in sub-images. The proposed method successfully counts overlapping erythrocytes and determines the cell type with an accuracy rate of 99.73%. The results of the proposed method have shown that it can be used efficiently in various fields.

KEYWORDS
Blood cells
Deep learning
Microscopy
Machine learning
Classification

INTRODUCTION

Developing fast and reliable methodologies and equipment that solve problems in the field of health provides various advantages to clinical staff and positively affects the quality of life of societies. Eliminating this developed equipment's cost and negative aspects is as essential as the development phase. Microscopes used in the laboratory are the most common example of this type of equipment. However, detecting and counting objects in complex images or videos from microscopes is a challenging and time-consuming task that is encountered in many traditional applications (Aliyu 2017). The reliability of the applications, especially in the medical field, is critical in the accurate and successful implementation of the diagnosis and treatment processes of the specialists. For this reason, systems that provide decision-support mechanisms to experts in the field are critical (Alimadadi *et al.* 2020).

Today, hematological tests, the standard procedure of every laboratory, are significant for the clinical diagnosis of cancer, anemia, and other blood diseases. However, analyzing the obtained data is a susceptible process (Ahn *et al.* 2018). Although there are various automated systems for performing these tests, one of the most used methods today is the examination of blood samples with peripheral smears under the microscope. Even today, this examination method is performed manually by trained hematol-

Manuscript received: 10 May 2022,
Revised: 27 June 2022,
Accepted: 28 June 2022.

[1] pala@subu.edu.tr (**Corresponding Author**)
[2] muratcimen@subu.edu.tr
[3] mustafayildiz@subu.edu.tr
[4] gcetinel@sakarya.edu.tr
[5] emiravcioglu@hitit.edu.tr
[6] yusufalaca@hitit.edu.tr

ogists. Experts obtain results by evaluating the morphological characteristics of blood cells, such as size and shape, by examining blood cells under a light microscope. This method, which has a high potential for obtaining erroneous results, requires effort (Bain 2005). One of the primary problems is the increase in morphological diversity due to intracellular and intercellular variations producing false results. The cells can be spread on the slide with concentric or edge overlapping in the peripheral spreading process. In addition, the variable properties of the image, such as color and contrast caused by the imaging system, cause differences between the same samples. In addition to all these physical factors, the skill and experience of the hematologist examining the sample also creates a subjective evaluation of the results. Quantitative analysis methods are gaining importance in overcoming these problems (Mohammed *et al.* 2014).

Blood consists of two separate components: plasma and blood cells. In the samples prepared by peripheral smear, three different types of cells can be observed: red blood cells (RBC), white blood cells (WBC), and platelets. Healthy adult individuals typically have $4,500 \times 10^3/\mu L$ RBCs, $8 \times 10^3/\mu L$ WBCs, and $300 \times 10^3/\mu L$ platelets. Platelets are the smallest cells of the blood, with an average diameter of $2-4\mu m$, disc-shaped morphology, and no nuclei. Granules in platelets contain substances that will instigate clot formation activity in case of bleeding. Therefore, the main task of platelets are hemostasis and prevention and control of bleeding. White blood cells, also known as leukocytes, have significant morphological differences compared to other blood cells and have a diameter of $10-20\mu m$. White blood cells do not contain hemoglobin, and the cytoplasm density is low, while the nuclear densities are high. An essential part of the immune system, Leukocytes move from the blood fluid to the tissues and protect the body. It counteracts damage by deactivating bacteria, viruses, or other foreign organisms and provides a defense mechanism by producing antibodies (Beydoun *et al.* 2016). Therefore, the total leukocyte concentration in the blood is a vital indicator of the human immune system, and many diseases can be detected only by calculating the leukocyte count. There are five different types of leukocytes: eosinophils, lymphocytes, neutrophils, monocytes, and basophils. Red blood cells, also known as erythrocytes, are the most common cell type in the blood, with a diameter of $7-10\mu m$ and a biconcave disc-shaped erythrocyte with a thickness of $2.2\mu m$ (Aliyu 2017). Their main functions in the body are to transmit oxygen to the tissues in circulation and to remove wastes and carbon dioxide from the tissues. Their color is red because they contain hemoglobin protein.

Researchers are making significant efforts to count the cells in the images obtained from a peripheral smear. As a result of these efforts, systems that produce results with remarkable accuracy have emerged. These traditional systems produce results by applying the steps of pre-processing, segmentation, feature extraction, and classification steps to the obtained images, respectively (Gonzalez *et al.* 2004). Pre-processing steps are used to remove the images' noise and the colors' distortions. In general, images are corrected by applying spatial or frequency plane operations. Examples are pre-processing algorithms' histogram correction, average receiver, and median filter. The most optimal images should be obtained for the segmentation step. The next step, segmentation, plays the most crucial role in the defined system and significantly affects its accuracy. There are various traditional or hybrid algorithms for segmentation and many methods developed for pre-processing (Kibunja 2021; Gould *et al.* 2009; Li *et al.* 2017; Çimen *et al.* 2019). In the feature extraction steps, the image segments' morphology, color coefficient, or other descriptive properties are obtained from the segmentation process (Zhang *et al.* 2019). The defining features obtained in this step are essential as they reveal the success of the classification. The primary purposes of feature extraction are to ensure that the images taken as input are defined as fingerprints and to identify the numerical or vector quantities obtained as output (Nixon and Aguado 2019). Various feature extraction methods that are flexible and adaptively may be preferred to improve performance rates. Various algorithms such as artificial neural networks, support vector machines, Naive Bayes networks, linear discriminant analysis, and multilayer networks are used in order to use the property values obtained from blood cells for counting and classification (Bayat *et al.* 2018; Ye *et al.* 2004).

This study proposes a new CNN-based strategy to eliminate the overlap-induced counting problem in blood samples prepared with peripheral smear and determine the blood cell type. In the proposed method, the images of the entire peripheral blood slide were divided into sub-images using adaptive image processing techniques to identify the overlapping cells and cell types. Each cropped image was labeled by the number of overlapping cells with hematology expert opinion, thus providing ground truth data. In addition, white blood cells are labeled as a separate class. CNN was used to classify cells divided into sub-images as blocks from the original images. The proposed method achieved 99.73% accuracy in counting overlapping red blood cells and separating RBC-WBC blood cell types. The results show that the proposed method can be adapted to areas where high-resolution images are found and reliable results.

PROPOSED METHOD

With the increased processing capacity of graphics processor units in recent years, deep learning methods have started to be used frequently in classification, recognition, and detection tasks, especially in the medical field (McLeod and Ozcan 2016; Chiroma *et al.* 2019; Jang and Cho 2019). Considering the problems related to blood count, using deep learning methods in blood count and classification steps can bring many advantages. The first of these advantages is that high image quality input is not required for deep learning methods, so problems arising from physical conditions are avoided (Wang *et al.* 2015). Secondly, it does not need to perform feature extraction and segmentation operations outside the system due to its convolutional layers. It produces results by applying straightforward approaches to solving complex problems that require high processing power (Dodge and Karam 2016). A Convolutional Neural Network (CNN), which is defined in literature as a unique structure of deep learning methods, is widely used to classify blood images and solve problems in other fields (Rere *et al.* 2016; Xue *et al.* 2016; Pala *et al.* 2022).

Convolutional neural networks are a learning architecture inspired by the visual perception mechanism of living things. CNN methods follow end-to-end training metrology, eliminating the pre-processing steps of complex images. CNN techniques are more similar to biological neural networks than other machine learning methods due to the layers – and it works very effectively on displaced, scaled, bent, or deformed images (Choi *et al.* 2017; Sun *et al.* 2019). The convolutional layer in the CNN architecture aims to learn the basic parameters that represent the features in the input images. Convolutional layers consist of various filters that allow the learning of different properties. These filters have various magnitudes and shift coefficients and are convoluted with the input image. As a result, each image taken as input is processed and given as output as a new feature image. By applying more than one convolution operation to the input layer, the depth of the network is increased, and with the network, more accurate results can be produced. The new feature layers obtained as the output of the convolutional layer can be high-valued compared to the inputs due to the multiplication process (Xue *et al.* 2016). This situation can cause overfitting in the network structure. In order to prevent this situation and increase the training performance, normalization layers can be added to the network structure (Huang *et al.* 2019). The main task of the normalization layer is to bring the values formed as a result of multiplications to a specific range and transmit the appropriate values to the next layer. Since images have a static structure, distinctive features found at one point in the image can also be found in other areas. This feature makes it possible to express the defining features of images with smaller areas. The pooling layers in the CNN structure enable these features that spread over large areas to be expressed in small areas. Pooling layers filter the input images with a specific size, like convolution layers. The pooling layer prevents situations such as memorization that will occur in the network structure. The outputs of the pooling layer are smaller than the input image, with the size depending on the filter size (Barbastathis *et al.* 2019; Liu *et al.* 2019; Strumberger *et al.* 2019).

Many high-performance methods have been proposed to classify non-overlapping blood cells. However, the overlap problem in blood cells counted using the peripheral smear method is widespread, and these processes are ignored when creating data sets. This is most commonly seen in the count of erythrocytes. Counting problems occur when at least two or more erythrocytes overlap. In this study, CNN was used to count the erythrocytes with overlapping observed in the samples prepared by the peripheral smear method and simultaneously make the RBC-WBC classification. The steps in this study are shown in Fig. 1.

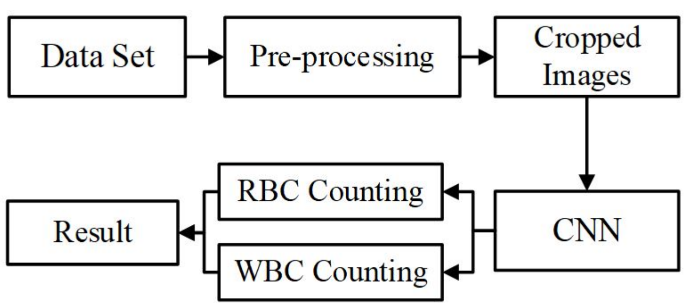


Figure 1 A brief schema of the proposed method

Data Set

LISC dataset was used in this study (Rezatofighi and Soltanian-Zadeh 2011). The LISC database was digitized in the hematology laboratory by preparing the blood samples collected from healthy individuals by the peripheral smear method. Different peripheral smear slides were prepared from 8 individuals, and 117 whole slide images were collected. The Gismo-Wright staining technique was used in the peripheral smear, and a microscope with a 100x optical lens was used to collect images. The obtained images were transferred to digital media using a camera. The images in the dataset are 720×576 resolution.

Proposed Cell Localization Method

A simple and adaptive pre-processing algorithm is intended to separate peripheral slide images into sub-images. Pre-processing step aims to divide the overlapping cells into sub-images and input them into the deep learning model more effectively. This way, the deep learning-based segmentation problem, which requires high computational operations, has been transferred to the classification problem. All the applied pre-processing steps are adaptive and can be used in applications such as real-time and various other datasets. Applying this pre-processing step to images allows for higher resolution blood image classification. In addition, giving the data as input to the CNN model by dividing it into sub-images provides computational efficiency during training and testing. First, the images in the RBC color space in the data set were converted to gray-level images. Gray-level 256-bit images were converted to binary images adaptively using the Otsu method (Otsu 1979). In binary-level images, the centers of erythrocytes and leukocytes resemble the background due to the cytoplasm structure. Therefore, the holes-filled method was applied to the centers of the obtained binary level images. While determining the center of the images at the binary level, the morphological erosion operator is applied to ensure the clarity of the edges. Finally, the centers of the blocks in the binary images were found. The bounding boxes' positions were mapped onto the RGB images in the original dataset. Bounding boxes in blocks were cropped from the original images, and a window size of 128×128 was transferred. The applied pre-processing steps are shown in Figure 2.

In Figure 3, the results of the applied pre-processing steps used to separate images into sub-images are given. RGB images were used during the training and testing times, and the pre-processing steps were used only to divide into sub-images.

Figure 4 shows images of randomly selected overlapping erythrocytes from the cropped sub-images.

Different numbers of erythrocytes overlap in the sub-images recorded due to pre-processing from the original dataset. The cropped sub-images were counted and labeled with the opinion of the hematologist. The number of overlapping cells in each image is labeled, and all WBC cells are labeled as a separate class. 117 RGB images with 720×576 resolution were found from the dataset, and 13345 RGB images with 128x128 sub-images were cropped and labeled. All sub-images were used during CNN training and testing. The distribution of the labels resulting from sub-images is given in Table 1.

Proposed CNN Model

In the proposed model, the images taken as input data during the training phase are forwarded to each layer, and then the model produces a result for each image. The loss function calculated the difference between the results obtained as the model output and the actual results. As a result of the loss obtained from each image, the model updates its internal weights to increase the learning rate. In the training phase of the model, the gradient descent optimization algorithms were used to update the optimum internal weights. After the training phase was completed, test images were used to test the model's success.

The general structure of the proposed CNN model is shown in Figure 5. In our proposed CNN model, we use four convolutional layers, three max-pooling layers, and two fully connected layers, which means two hidden layers and one output layer. $128 \times 128 \times 3$ sub-image from the data set normalized and applied to the model as input. To construct the 1st, 2nd, 3rd, and 4th convolutional layers, we used filter sizes of 3x3, 2x2, 3×3, and 3×3 and stride sizes of all layers are 1×1. The ReLU activation function and the same padding technique are used in all convolution layers. 2x2 kernel max-pooling with 2×2 stride is applied for the mapping

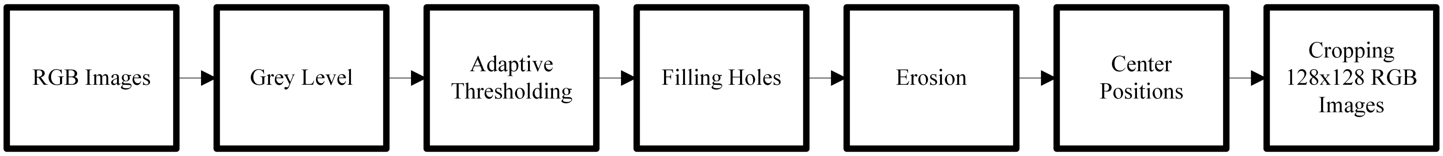


Figure 2 Applied pre-processing steps

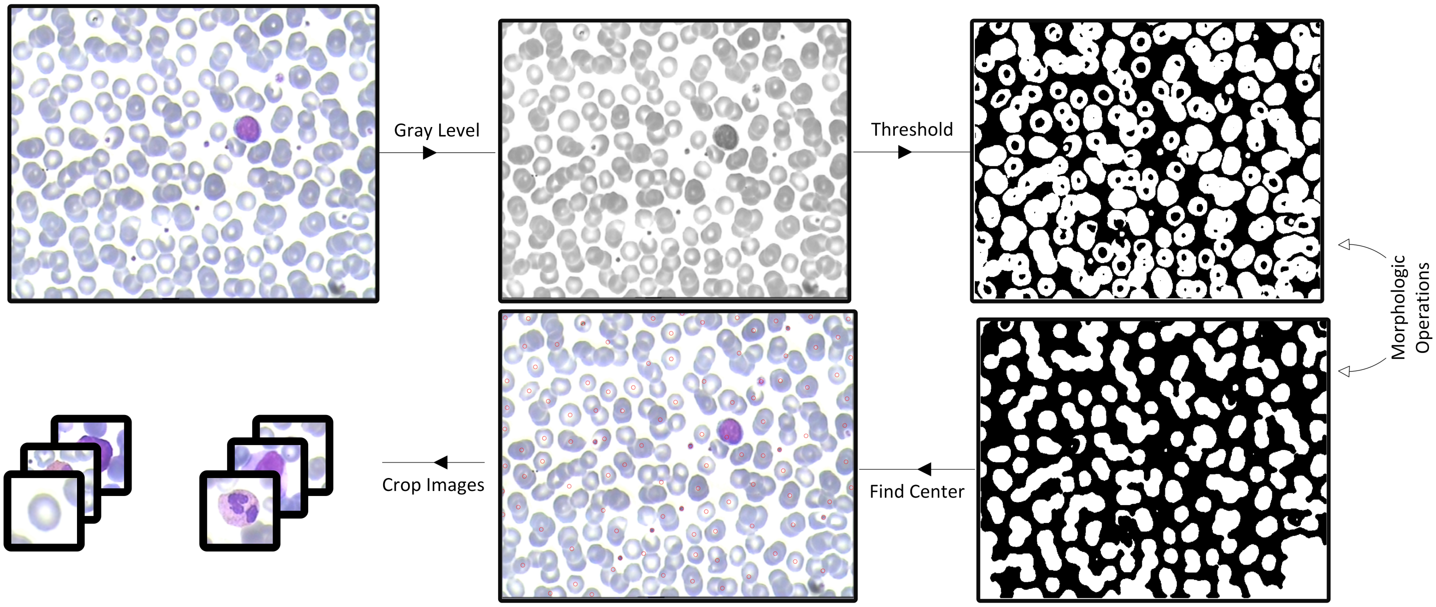


Figure 3 Cropping images into sub-images with pre-processing steps

Table 1 Data distribution

	Number of overlapped RBCs									WBC	
Class	1	2	3	4	5	6	7	8	9	10	Total
Number of samples	2545	1041	2040	2779	2553	1346	422	113	155	351	13345

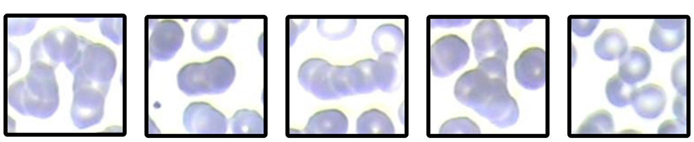


Figure 4 Examples of erythrocytes with different numbers overlapping

feature. A dropout layer of 0.25 was used at the output of all pooling layers. ReLU activation function is used in all convolution layers. The maximum-pooling layer's output matrix in the second block is flattened and transferred to the dense block. There is a neural network in the last layer for classifying the number and type of blood samples. The fully connected layer consists of two dense layers of 512 and 256 neurons and an output layer of 10. In order to prevent overfitting, 0.2 dropout was used in the second dense layer. The dropout rate is chosen experimentally. We use the softmax activation function in the output layer. The Adam optimization function is used in the training steps, and the batch size is 64. Early stopping was used depending on the accuracy rate during the proposed CNN model training. The best weights obtained during the training process were saved.

Experimental Results

Various performance criteria were used to test the performance of the proposed CNN model. Early stopping was used depending on the accuracy during the proposed CNN model training. The training was stopped at the 50 epochs when the accuracy and loss stabilized. The weights with the highest accuracy and lowest loss value were recorded. Figure 6 shows the accuracy and loss graphs of the training and validation results. The proposed CNN model reached the highest accuracy value of 99.73% in 50 epochs. The hardware environment includes an Intel Core i7-7700 HQ CPU, 16 GB of RAM, and the Windows 10 operating system (64-bit mode) to implement the model. In order to accelerate the computations and improve efficiency, GPU-accelerated computing with NVIDIA GTX 1050 is also utilized.

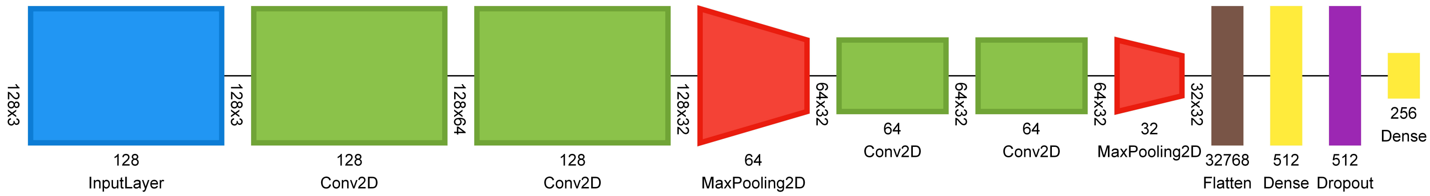
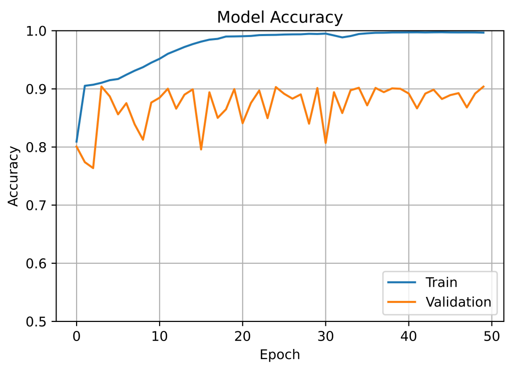
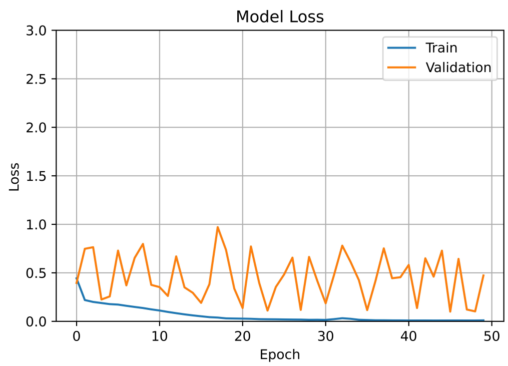


Figure 5 Proposed CNN model



(a) Accuracy of train and validation



(b) Loss of train and validation

Figure 6 Results of the proposed CNN model performance results

Classification results were evaluated using other performance criteria such as Precision, Recall, and F1-Score. Precision, Recall, and F1-Score in the proposed CNN model were 93.42%, 96.27%, and 94.73%, respectively. In addition, the confusion matrix was used to show the success of the proposed method in the test data. The confusion matrix shows the correct and incorrect classification of the test data of the proposed method in detail. The confusion matrix of the proposed model is shown in Figure 7.

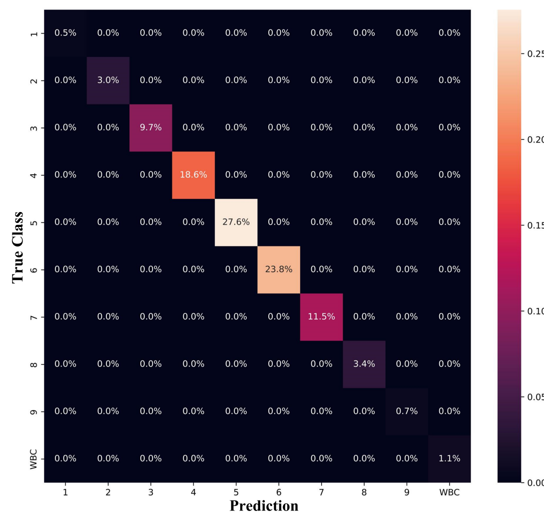


Figure 7 The confusion matrix of the proposed CNN model

In order to count the overlapping cells, the segmentation problem was transformed into a classification problem. In addition, giving the data as input to the CNN model by dividing it into sub-images provides computational efficiency during training and testing. In the proposed method, peripheral blood images were divided into sub-images, block by block, using adaptive image processing techniques to identify overlapping cells and cell types. Each image was labeled with an expert opinion. Each overlap number in red blood cells is labeled as a separate class. In addition, white blood cells are labeled as a separate class. Data augmentation methods were applied to ensure data distribution. Convolutional neural networks were used for classification. Convolutional neural networks can learn the properties of erythrocytes and leukocytes with more than one overlap in the image, thus producing more successful results than the methods used. The proposed CNN model consists of three blocks and has significantly less computational complexity. Classifier performances are measured using accuracy, precision, recall, and F1-Score metrics. Validation of results has been carried out by tenfold cross-validation. CNN-based approach counts overlapping cells and determines cell type with 99.73% accuracy. Precision, Recall, and F1-Score in the proposed CNN model were 93.42%, 96.27%, and 94.73%, respectively. It is seen that the proposed method performs multitasking with higher accuracy compared to other methods.

CONCLUSION

Accurate counting and classifying of blood samples are critical in diagnosing diseases and following treatments. The peripheral smear method is the most commonly used method in laboratories and clinics for cell counting and determining type. However, overlapping erythrocytes is one of the biggest obstacles to a reliable counting process. This study proposes a new CNN-based strategy to eliminate the overlap-induced counting problem in peripheral smear blood samples and accurately determine the blood cell type.

Acknowledgments

This work was supported by Sakarya University of Applied Science Scientific Research Projects Coordination Unit (SUBU BAPK, Project Number: 2020-01-01-011). The author, Muhammed Ali PALA, is grateful to The Scientific and Technological Research Council of Turkey for granting a scholarship (TUBITAK, 2211C, Grant Number: 1649B032100653) for his Ph.D. studies.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Availability of data and material

Previously reported data were used this study and are available at (Rezatofighi and Soltanian-Zadeh 2011). These prior studies (and datasets) are cited as references at relevant places within the text.

LITERATURE CITED

Ahn, D., J. Lee, S. Moon, and T. Park, 2018 Human-level blood cell counting on lens-free shadow images exploiting deep neural networks. Analyst **143**: 5380–5387.

Alimadadi, A., S. Aryal, I. Manandhar, P. B. Munroe, B. Joe, *et al.*, 2020 Artificial intelligence and machine learning to fight covid-19.

Aliyu, H. A., 2017 Detection of accurate segmentation in blood cells count–a review. International Journal of Science & Engineering Development Research **2**: 28–32.

Bain, B. J., 2005 Diagnosis from the blood smear. New England Journal of Medicine **353**: 498–507.

Barbastathis, G., A. Ozcan, and G. Situ, 2019 On the use of deep learning for computational imaging. Optica **6**: 921–943.

Bayat, F. M., M. Prezioso, B. Chakrabarti, H. Nili, I. Kataeva, *et al.*, 2018 Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits. Nature communications **9**: 1–7.

Beydoun, M., H. Beydoun, G. Dore, J. Canas, M. Fanelli-Kuczmarski, *et al.*, 2016 White blood cell inflammatory markers are associated with depressive symptoms in a longitudinal study of urban adults. Translational psychiatry **6**: e895–e895.

Chiroma, H., A. Y. Gital, N. Rana, S. M. Abdulhamid, A. N. Muhammad, *et al.*, 2019 Nature inspired meta-heuristic algorithms for deep learning: recent progress and novel perspective. In *Science and Information Conference*, pp. 59–70, Springer.

Choi, J. W., Y. Ku, B. W. Yoo, J.-A. Kim, D. S. Lee, *et al.*, 2017 White blood cell differential count of maturation stages in bone marrow smear using dual-stage convolutional neural networks. PloS one **12**: e0189259.

Çimen, M. E., Z. GARİP, M. A. PALA, A. F. BOZ, and A. AKGÜL, 2019 Modelling of a chaotic system motion in video with artificial neural networks. Chaos Theory and Applications **1**: 38–50.

Dodge, S. and L. Karam, 2016 Understanding how image quality affects deep neural networks. In *2016 eighth international conference on quality of multimedia experience (QoMEX)*, pp. 1–6, IEEE.

Gonzalez, R. C., S. L. Eddins, and R. E. Woods, 2004 *Digital image publishing using MATLAB*. Prentice Hall.

Gould, S., T. Gao, and D. Koller, 2009 Region-based segmentation and object detection. Advances in neural information processing systems **22**.

Huang, X., Q. Zhang, G. Wang, X. Guo, and Z. Li, 2019 Medical image super-resolution based on the generative adversarial network. In *Chinese Intelligent Systems Conference*, pp. 243–253, Springer.

Jang, H.-J. and K.-O. Cho, 2019 Applications of deep learning for the analysis of medical data. Archives of pharmacal research **42**: 492–504.

Kibunja, K. P., 2021 *A Duplicate number plate detection system using fixed location cameras*. Ph.D. thesis, Strathmore University.

Li, X.-W., Y.-X. Kang, Y.-L. Zhu, G. Zheng, and J.-D. Wang, 2017 An improved medical image segmentation algorithm based on clustering techniques. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–5, IEEE.

Liu, T., K. De Haan, Y. Rivenson, Z. Wei, X. Zeng, *et al.*, 2019 Deep learning-based super-resolution in coherent imaging systems. Scientific reports **9**: 1–13.

McLeod, E. and A. Ozcan, 2016 Unconventional methods of imaging: computational microscopy and compact implementations. Reports on Progress in Physics **79**: 076001.

Mohammed, E. A., M. M. Mohamed, B. H. Far, and C. Naugler, 2014 Peripheral blood smear image analysis: A comprehensive review. Journal of pathology informatics **5**: 9.

Nixon, M. and A. Aguado, 2019 *Feature extraction and image processing for computer vision*. Academic press.

Otsu, N., 1979 A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics **9**: 62–66.

Pala, M. A., M. E. Çimen, A. Akgül, M. Z. Yıldız, and A. F. Boz, 2022 Fractal dimension-based viability analysis of cancer cell lines in lens-free holographic microscopy via machine learning. The European Physical Journal Special Topics **231**: 1023–1034.

Rere, L., M. I. Fanany, and A. M. Arymurthy, 2016 Metaheuristic algorithms for convolution neural network. Computational intelligence and neuroscience **2016**.

Rezatofighi, S. H. and H. Soltanian-Zadeh, 2011 Automatic recognition of five types of white blood cells in peripheral blood. Computerized Medical Imaging and Graphics **35**: 333–343.

Strumberger, I., E. Tuba, N. Bacanin, R. Jovanovic, and M. Tuba, 2019 Convolutional neural network architecture design by the tree growth algorithm framework. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE.

Sun, Y., B. Xue, M. Zhang, and G. G. Yen, 2019 Completely automated cnn architecture design based on blocks. IEEE transactions on neural networks and learning systems **31**: 1242–1254.

Wang, H., K. Sun, and S. He, 2015 Characteristic analysis and dsp realization of fractional-order simplified lorenz system based on adomian decomposition method. International Journal of Bifurcation and Chaos **25**: 1550085.

Xue, Y., N. Ray, J. Hugh, and G. Bigras, 2016 Cell counting by regression using convolutional neural network. In *European Conference on Computer Vision*, pp. 274–290, Springer.

Ye, J., R. Janardan, and Q. Li, 2004 Two-dimensional linear discriminant analysis. Advances in neural information processing systems **17**.

Zhang, Y., T.-C. Poon, P. W. Tsang, R. Wang, and L. Wang, 2019 Review on feature extraction for 3-d incoherent image processing using optical scanning holography. IEEE Transactions on Industrial Informatics **15**: 6146–6154.

How to cite this article: Pala, M. A., Cimen, M. E., Yildiz, M. Z., Cetinel, G., Avcioglu, E., and Alaca, Y. CNN-Based Approach for Overlapping Erythrocyte Counting and Cell Type Classification in Peripheral Blood Images. Chaos Theory and Applications, 4(2), 82-87, 2022.

CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.4 / No.2 / 2022 / pp.88-93
https://doi.org/10.51537/chaos.1101581

The FPGA-Based Realization of the Electromagnetic Effect Defined FitzHugh-Nagumo Neuron Model

Nimet Korkmaz *,1 **and Bekir Şıvga** *,2
*Kayseri University, Department of the Electrical and Electronics Engineering, Talas, 38280, Kayseri, Turkey.

ABSTRACT The electrical transmission, which occurs on the surface of the neuron membranes, is based on the flow of charges such as calcium, potassium and sodium. This potential change means a current flow and if there is a variable current flow, a flux change comes into question. Accordingly, recent studies have suggested that these electrophysiological neuronal activities can induce a time-varying electromagnetic field distribution. The electric field is usually defined as an external stimulation variable of the biological neuron models in literature. However, the electric field is included in the biological neuron models as a new state variable in another perspective and it is described the polarization modulation of media. Here, this study focused on that the electric field is a state variable in the biological neuron model. The numerical simulations of the FitzHugh-Nagumo neuron, which is improved by including the electromagnetic effect, are re-executed in this study. Then, the hardware realization of this system is built on the FPGA device. Therefore, it is shown that it is also possible to perform the hardware realizations of the neuronal systems, which have a new state variable for the electric field definition.

KEYWORDS
Electromagnetic field
Biological neuron model
Hardware realization
Field Programmable Gate Array (FPGA)
FitzHugh-Nagumo

INTRODUCTION

The basic unit of the nervous system is neuron cells and it also constitutes the main unit of the communication system, which is based on electrical transmission, in the living beings. The electrical transmission, which occurs on the surface of the neuron membranes, is based on the flow of charges such as calcium, potassium and sodium. This potential change means a current flow and if there is a variable current flow, a flux change comes into question [Ma and Tang 2015; Lv *et al.* 2016; Lv and Ma 2016; Wu *et al.* 2017; Xu *et al.* 2017; Ma *et al.* 2017; Ge *et al.* 2018]. Accordingly, recent studies have suggested that these electrophysiological neuronal activities can induce a time-varying electromagnetic field distribution.

In [Lv *et al.* 2016], the expression magnetic flux is associated with a memristor element. The membrane potential definition is combined with the memristor definition. Thus, it has been suggested that magnetic flux can be used to explain the effect of electromagnetic induction. Based on the result in [Lv *et al.* 2016], different studies have also been put forward. For example in [Wu *et al.* 2016], the electromagnetic radiation has been considered as an external stimuli of the FitzHugh-Nagumo model and it have been observed the electrical activates of the neuron by relating with the sudden heart disorder under the heavily electromagnetic radiation. An improved cardiac model has been exposed to an external electromagnetic radiation in [Ma *et al.* 2017] and it has been founded that the electromagnetic radiation causes the quiescent state of the membrane potentials.

When the electric field distribution induced by the fluctuations in the action potential becomes apparent, this effect has been included in the definitions of the biological neuron models. For example, in [Bao *et al.* 2018], Hindmarsh–Rose neuron model has been modified by utilizing the memristor device characteristic. It has been thought to observe neuronal dynamics under the electromagnetic induction and these studies have been confirmed by the circuit breadboard based experimental results. In [Bao *et al.* 2019], an electromagnetic induction current has been generated by the threshold memristor. This current has been applied to Hindmarsh-Rose neuron model instead of the external current definition. This neuron model has been presented as a memristive defined system and this system has been implemented with discrete device on hardware breadboards for validating electronic neuron.

A locally active memristive defined neuron model has been proposed by using the FitzHugh-Nagumo neuron model in [Lin *et al.*

Manuscript received: 11 April 2022,
Revised: 31 May 2022,
Accepted: 8 July 2022.

[1] nimetkorkmaz@kayseri.edu.tr (**Corresponding Author**)
[2] bekir.sivga@saglik.gov.tr

2020]. The firing patterns and multistability of this neuronal system have been investigated and these systems have been realized by emulating the memristor definition with the analog electronic elements. In [Bao *et al.* 2021], a memristive neuron model with the adapting synapse has been imitated by a flux controlled memristor and a memristive mono-neuron model has been implemented by a fitting activation function circuit.

The electric field has been defined as an external stimulation variable of the biological neuron models in the outlined systems in above. However, the electric field has been included in the biological neuron models as a new state variable in [Ma *et al.* 2019] and it has been described the polarization modulation of media that is resulted from the external electric field or the intrinsic change of ions on the membrane surface in mentioned study. Inspiring by [Ma *et al.* 2019], the numerical simulations of the FitzHugh-Nagumo neuron, which is improved by including the electromagnetic effect, are re-executed in this study. Then, the hardware realization of this system has been built on the FPGA device for the first time. Therefore, it is shown that it is also possible to perform the hardware realizations of the neuronal systems, which have a new state variable for the electric field definition, similar to the memristive defined ones.

In this context, firstly after the introducing of the improved the FitzHugh-Nagumo neuron model by including the electromagnetic effect, the repeated numerical simulation results are given in Section 2. The FPGA-based hardware implementation stages of the relevant model and its obtained experimental results are presented in Section 3. The outputs of this study are discussed in the last section.

THE INVESTIGATING OF THE ELECTROMAGNETIC EFFECT DEFINED FITZHUGH-NAGUMO NEURON MODEL

The electrical transmission, which occurs on the surface of the neuron membranes, is based on the flow of charges such as calcium, potassium and sodium. This potential change means a current flow and if there is a variable current flow, a flux change comes into question [Ma and Tang 2015; Lv *et al.* 2016; Lv and Ma 2016; Wu *et al.* 2017; Xu *et al.* 2017; Ma *et al.* 2017; Ge *et al.* 2018]. Accordingly, recent studies have suggested that these electrophysiological neuronal activities can induce a time-varying electromagnetic field distribution. In order to describe this electromagnetic field distribution, the membrane surface has been considered as a charged plate and the charge density τ of its surface has been written as the ratio of the electrical charge 'q' to the surface area 'S' of the plate, namely.

When the dielectric constant is 'ϵ', the induced electromagnetic field has been given as '$E = (q/2S\epsilon)$' or '$E = (q/2\epsilon)$' for a sphere shape neuron with the 'r' radius. The voltage difference between the charged plates can be defined by depending on the electromagnetic field as in '$V = rE \cong E\sqrt{S}$'. In neurons, the lipid layers of the neuron membrane are considered as conductive material and the space between these layers as an insulating material, so a capacitor definition 'C' is usually added to the biological neuron models [Hodgkin and Huxley 1952; Morris and Lecar 1981]. Similarly, the induced electromagnetic field 'E' has been considered as a new state variable in [Ma *et al.* 2019] by taking into consideration the inductance 'L' of the media 'p'. This assumption is formulated in general terms as in Eq.1.

$$\begin{aligned} C\frac{dV}{dt} &= f(V,i,p) \\ L\frac{di}{dt} &= g(V,i) + rE \\ \frac{dE}{dt} &= \frac{1}{2S\epsilon}\frac{dq}{dt} = \frac{1}{2S\epsilon}i = ki \end{aligned} \tag{1}$$

In Eq.1, 'f' and 'g' functions are the nonlinear expressions and they represent the membrane potential and the transmembrane current. These features have been adapted to the FitzHugh–Nagumo neuron model and the electromagnetic effect defined FitzHugh-Nagumo neuron model has been improved as in Eq.2 [Ma *et al.* 2019].

$$\begin{aligned} \tau\frac{dx}{dt} &= x - \frac{x^3}{3} - y + I_{ext} \\ \frac{dy}{dt} &= ax + by + d + rE \\ \frac{dE}{dt} &= ky \end{aligned} \tag{2}$$

where, while the 'x' state variable describes activation of the membrane potential, the 'y' state variable represents the inactivation of the neuron. The 'a', 'b', 'd' and 'τ' are the model parameters. The external current stimulates are given by the 'I_{ext}' parameter. In biological neuron models that do not include the electromagnetic field effect, the external currents applied to the neurons are generally defined as the DC currents [Izhikevich 2003; Fitzhugh 1965; Hindmarsh and Rose 1984]. However, a flux change, namely a time-depended current, requires for seeing the effect of the electromagnetic field. Thus, the external current in the electromagnetic effect defined FitzHugh-Nagumo neuron model can be formed as a sinusoidal source. In fact, the amplitude and the frequency of this current affect the dynamical behaviors of the neuron model. The effect of the frequency on the dynamical behaviors of the electromagnetic effect defined FitzHugh-Nagumo neuron model have been observed via a bifurcation diagram in Figure 1 by fixing the amplitude to 0.1 in here.

The numerical simulation results re-executed for four different frequency values of the sinusoidal source are given in Figure 2, respectively. In these numerical simulations, the values of 'a', 'b', 'd' and 'τ' parameters have been chosen as follows: $a = 3, b = -3, d = 5, k = 15, r = 0.0001$ and $\tau = 0.1$.

THE FPGA-BASED REALIZATION PROCESS

Some specialties are desired in the electronic equipment that is used in the hardware realization studies of the bio-inspired systems. Some of these most preferred ones are low power and low device consumption, allowing different designs to be tested without the need for additional processes, and rapid prototyping. While there are studies in which biological neuron models are supported by the discrete device based hardware for providing advantages in terms of material supply and practical implementation in the literature [Sánchez-Sinencio and Linares-Barranco 1989; Linares-Barranco *et al.* 1991], the studies using programmable and reconfigurable analog/digital devices have also attracted attention in recent years [Korkmaz *et al.* 2016; Korkmaz and Kilic 2014; Karataş *et al.* 2022]. The programmable and re-configurable analog/digital devices combine many features mentioned above.

In this study, the Field Programmable Gate Array (FPGA) device is used for the hardware implementation of the electromagnetic effect defined FitzHugh-Nagumo neuron model. In addition

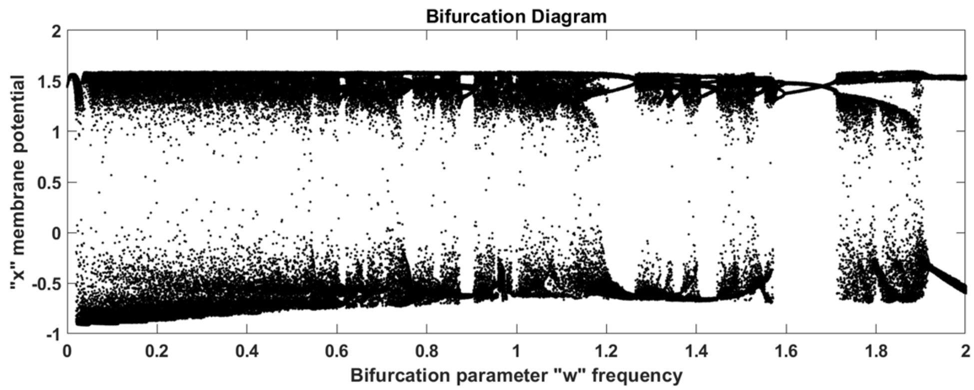


Figure 1 Bifurcation diagram of the membrane potential that is plotted by applying different frequencies in external stimulus within 200 time sample.

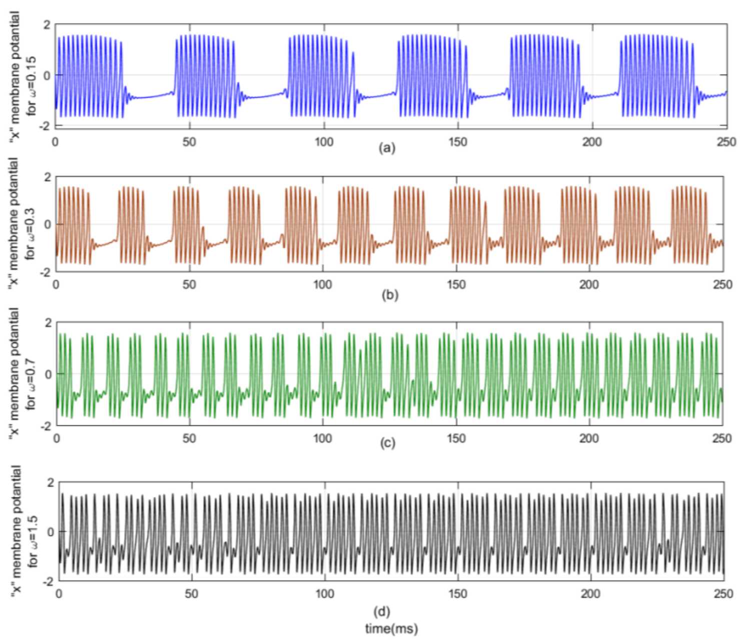


Figure 2 Numerical simulation results of the electromagnetic effect defined FitzHugh-Nagumo neuron model for different frequencies [a) 0.15, b) 0.3, c) 0.7 and d) 1.5] of the external stimulus.

to features mentioned above, the FPGA device is a digital electronic equipment operating with a parallel working procedure and having the programmability and reconfigurability features. The FPGA is preferred for the prototype realization of many models. Since the FPGA device is digital electronic equipment; the electromagnetic effect defined FitzHugh-Nagumo neuron model, which is defined by the ordinary differential equations, must be converted to a discrete-time expression for the FPGA-based implementation. Here, the Euler discretization method is used for this conversion process and the step size is set as $\Delta h = 0.01$. After applying the discretization method to the model in Eq.2, the obtained final definition is given in Eq.3.

$$\begin{aligned}
x_{i+1} &= [\frac{x_i - \frac{x_i^3}{3} - y + I_{ext}}{\tau}] * \Delta h + x_i \\
y_{i+1} &= [ax_i + by_i + d + rE_i] * \Delta h + y_i \\
E_{i+1} &= [ky_i] * \Delta h + x_i
\end{aligned} \tag{3}$$

The "System Generator for $DSP - XILINX^{TM}$" program tool is used for the FPGA-based implementation of the electromagnetic effect defined FitzHugh-Nagumo neuron model in Eq.3. This program provides an automatic conversion between the $MATLAB - SIMULINK^{TM}$ and the $XILINX^{TM}$ codes. After the conversion process, the system built on $MATLAB - SIMULINK^{TM}$ can be embedded into the FPGA device produced by $XILINX^{TM}$, directly [Xilinx July, 2022]. Figure 3 shows a diagram that is designed with the System Generator for DSP tool for the electromagnetic effect defined FitzHugh-Nagumo neuron model.

A multiplexer is added to this design for the selection of the frequency values as $\omega(rad/s) = [0.15, 0.3, 0.7, 1.5]$. Thus, instead of adjusting the frequency to four different values separately, the all frequency values are embedded to the FPGA in the same design. Thus, the implementation results could be easily observed by changing the switch positions on the FPGA development board.

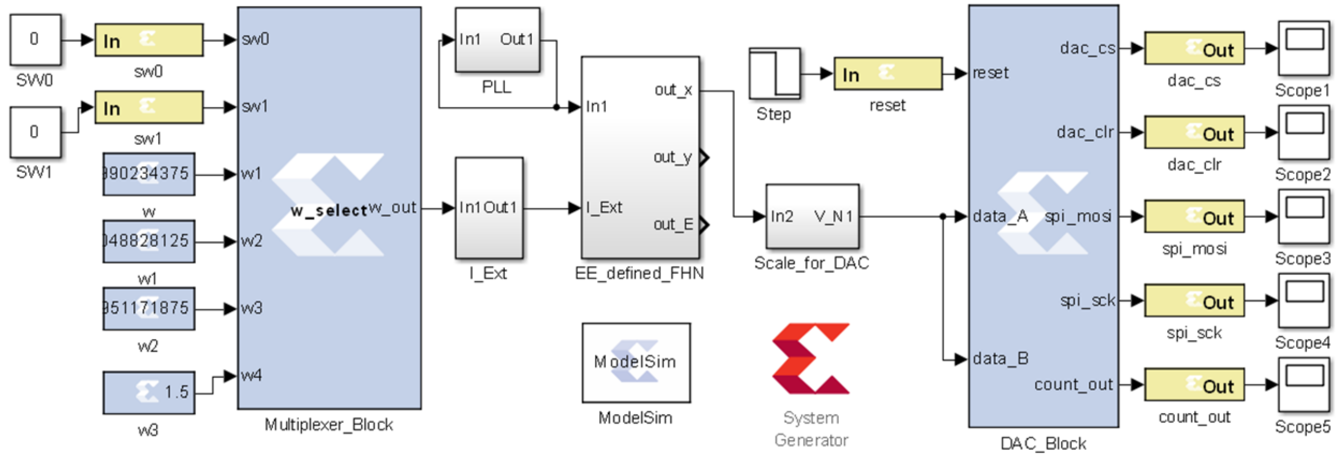


Figure 3 System Generator for DSP tool based design schema for the electromagnetic effect defined FitzHugh-Nagumo neuron model.

The electromagnetic effect defined FitzHugh-Nagumo neuron model has been built by using the predefined blocks in the System Generator for $DSP - XILINX^{TM}$ tool and it has been given by a subsystem illustrations named by "EE_defined_FHN" in Figure 3. The fixed-point arithmetic $Q = (32, 18)$ has been used in the design. After the automatic conversion process, the VHDL codes have embedded to the SPARTAN-3AN development board of $XILINX^{TM}$ company. A digital-to-analog converter (LTC2624) is available on this development board. The measurement results performed for the $\omega(rad/s) = [0.15, 0.3, 0.7, 1.5]$ frequency values have recorded by using the mentioned digital-analog converter. The FPGA-based realization results of the electromagnetic effect defined FitzHugh-Nagumo neuron model are given in Figure 4 for these frequencies. Therefore, it has been proved that it is also possible to perform the hardware realizations of the neuronal systems, which have a new state variable for the electric field definition.

This realization results are very similar to the obtained results for the numerical simulation in Figure 2. According to this similarity, the FPGA-based hardware implementation of the electromagnetic effect defined FitzHugh-Nagumo neuron model has been completed successfully. Some synthesis results of the FPGA-based realized model are presented in Table 1.

Table 1 Area usages and synthesis results in the FPGA-based implementation of the electromagnetic effect defined FitzHugh-Nagumo neuron model.

Area Usages Name	Area Usages Rate%
The used amount from 11776 REGISTER	184 (1%)
The used amount from 11776 4-INPUT LUT	1825 (15%)
The used amount from 5888 SLICE	975 (16%)
The used amount from 24 BUFGMUX	2 (8%)
The used amount from 20 MULT18X18SIO	19 (95%)
Maximum Delay (ns)	1083

CONCLUSION

In this study, the FPGA-based hardware realization of the electromagnetic effect defined FitzHugh-Nagumo neuron model has been handled. This biological neuron model stands out in terms of explaining the effect of the electric field on the neuron with a new state variable. In this context, after the investigating of the electromagnetic effect defined FitzHugh-Nagumo neuron model briefly, a bifurcation diagram has been plotted to observe the effect of the external time-depended sources on the dynamical behaviors of this neuron model. Numerical simulation studies have been carried out to observe neuron dynamics for different angular frequency values. Then, in order to demonstrate the adaptability of this biological definition to an electronically platform, the electromagnetic effect defined FitzHugh-Nagumo neuron model has been implemented with the FPGA device.

In the model definition, a time-depended current requires for seeing the effect of the electromagnetic field and a sinusoidal signal has been included in the model definition. In this study, this sinusoidal external signal in continuous time has been constructed on the FPGA device without requiring the usage of a LUT block or an external ADC. The System Generator for DSP tool has been used in this implementation process. The recorded experimental results show that an electromagnetic effect defined FitzHugh-Nagumo neuron model can be implemented with FPGA device, successfully.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Availability of data and material

Not applicable.

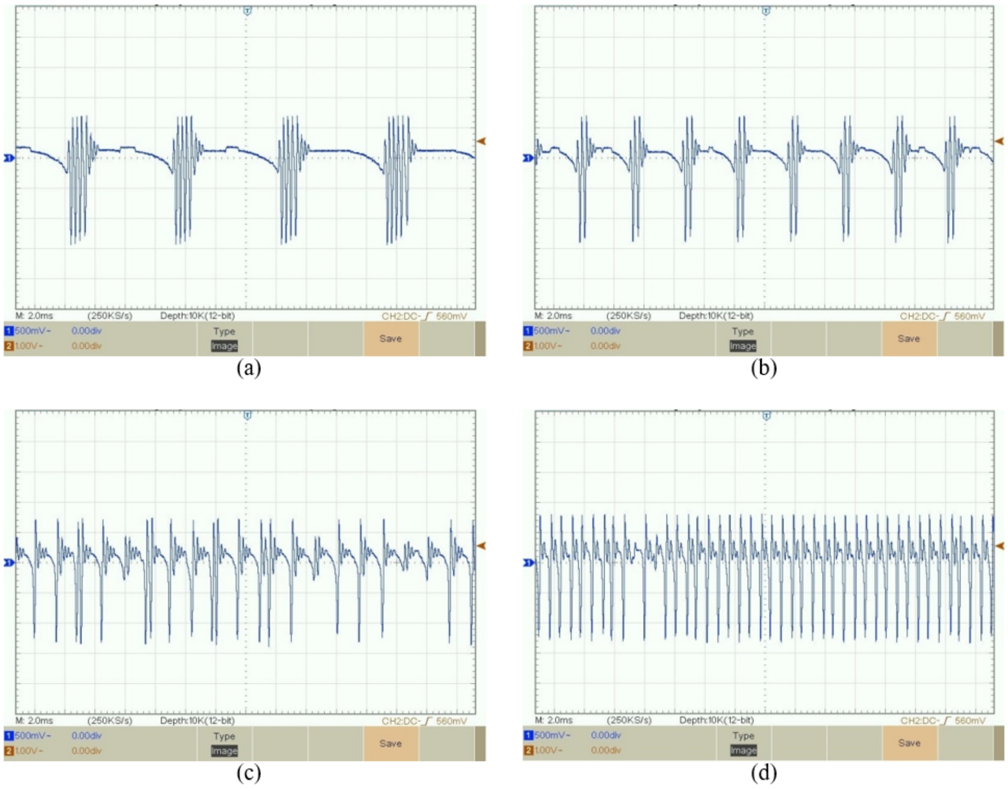


Figure 4 FPGA-based experimental realization results of the electromagnetic effect defined FitzHugh-Nagumo neuron model for different frequencies [a) 0.15, b) 0.3, c) 0.7 and d) 1.5] of the external stimulus.

LITERATURE CITED

Bao, B., A. Hu, H. Bao, Q. Xu, M. Chen, *et al.*, 2018 Three-dimensional memristive hindmarsh–rose neuron model with hidden coexisting asymmetric behaviors. Complexity **2018**.

Bao, B., Y. Zhu, J. Ma, H. Bao, H. Wu, *et al.*, 2021 Memristive neuron model with an adapting synapse and its hardware experiments. Science China Technological Sciences **64**: 1107–1117.

Bao, H., A. Hu, W. Liu, and B. Bao, 2019 Hidden bursting firings and bifurcation mechanisms in memristive neuron model with threshold electromagnetic induction. IEEE transactions on neural networks and learning systems **31**: 502–511.

Fitzhugh, R., 1965 Mathematical models of excitation and propagation nerve. Biological Engineering .

Ge, M., Y. Jia, Y. Xu, and L. Yang, 2018 Mode transition in electrical activities of neuron driven by high and low frequency stimulus in the presence of electromagnetic induction and radiation. Nonlinear Dynamics **91**: 515–523.

Hindmarsh, J. L. and R. Rose, 1984 A model of neuronal bursting using three coupled first order differential equations. Proceedings of the Royal society of London. Series B. Biological sciences **221**: 87–102.

Hodgkin, A. L. and A. F. Huxley, 1952 A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of physiology **117**: 500.

Izhikevich, E. M., 2003 Simple model of spiking neurons. IEEE Transactions on neural networks **14**: 1569–1572.

Karataş, F., İ. Koyuncu, M. Tuna, M. Alçın, E. Avcioglu, *et al.*, 2022 Design and implementation of arrhythmic ecg signals for biomedical engineering applications on fpga. The European Physical Journal Special Topics **231**: 869–884.

Korkmaz, N. and R. Kilic, 2014 Implementations of modified chaotic neural models with analog reconfigurable hardware. International Journal of Bifurcation and Chaos **24**: 1450046.

Korkmaz, N., İ. ÖZTÜRK, and R. Kilic, 2016 Multiple perspectives on the hardware implementations of biological neuron models and programmable design aspects. Turkish Journal of Electrical Engineering and Computer Sciences **24**: 1729–1746.

Lin, H., C. Wang, Y. Sun, and W. Yao, 2020 Firing multistability in a locally active memristive neuron model. Nonlinear Dynamics **100**: 3667–3683.

Linares-Barranco, B., E. Sánchez-Sinencio, Á. Rodríguez-Vazquez, and J. L. Huertas, 1991 A cmos implementation of fitzhugh-nagumo neuron model. IEEE Journal of Solid-State Circuits **26**: 956–965.

Lv, M. and J. Ma, 2016 Multiple modes of electrical activities in a new neuron model under electromagnetic radiation. Neurocomputing **205**: 375–381.

Lv, M., C. Wang, G. Ren, J. Ma, and X. Song, 2016 Model of electrical activity in a neuron under magnetic flow effect. Nonlinear Dynamics **85**: 1479–1490.

Ma, J. and J. Tang, 2015 A review for dynamics of collective behaviors of network of neurons. Science China Technological Sciences **58**: 2038–2045.

Ma, J., F. Wu, and C. Wang, 2017 Synchronization behaviors of coupled neurons under electromagnetic radiation. International Journal of Modern Physics B **31**: 1650251.

Ma, J., G. Zhang, T. Hayat, and G. Ren, 2019 Model electrical activity of neuron under electric field. Nonlinear dynamics **95**: 1585–1598.

Morris, C. and H. Lecar, 1981 Voltage oscillations in the barnacle giant muscle fiber. Biophysical journal **35**: 193–213.

Sánchez-Sinencio, E. and B. Linares-Barranco, 1989 Circuit implementation of neural fitzhugh-nagumo equations. In *Proceedings of the 32nd Midwest Symposium on Circuits and Systems,*, pp. 244–247, IEEE.

Wu, F., C. Wang, W. Jin, and J. Ma, 2017 Dynamical responses in a new neuron model subjected to electromagnetic induction and phase noise. Physica A: Statistical Mechanics and its Applications **469**: 81–88.

Wu, F., C. Wang, Y. Xu, and J. Ma, 2016 Model of electrical activity in cardiac tissue under electromagnetic induction. Scientific reports **6**: 1–12.

Xilinx, July, 2022 Xilinx inc. http://www.xilinx.com.

Xu, Y., H. Ying, Y. Jia, J. Ma, and T. Hayat, 2017 Autaptic regulation of electrical activities in neuron under electromagnetic induction. Scientific Reports **7**: 1–12.

How to cite this article: Korkmaz, N., and Şıvga, B. The FPGA-Based Realization of the Electromagnetic Effect Defined FitzHugh-Nagumo Neuron Model. Chaos Theory and Applications, 4(2), 88-93, 2022.

CHAOS Theory and Applications in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.4 / **No.**2 / 2022 / **pp.**94-103
https://doi.org/10.51537/chaos.1116084

On the Prediction of Chaotic Time Series using Neural Networks

Josué Alexis Martínez-García [*,1], **Astrid Maritza González-Zapata** [α,2], **Ericka Janet Rechy-Ramírez** [*,3] **and Esteban Tlelo-Cuautle** [α,4]

[*] University of Veracruz, Artificial Intelligence Research Institute, 91907, Veracruz, Mexico, [α] Instituto Nacional de Astrofisica, Optica y Electronica, Electronics Department, 72840, Puebla, Mexico.

ABSTRACT Prediction techniques have the challenge of guaranteeing large horizons for chaotic time series. For instance, this paper shows that the majority of techniques can predict one step ahead with relatively low root-mean-square error (RMSE) and Symmetric Mean Absolute Percentage Error (SMAPE). However, some techniques based on neural networks can predict more steps with similar RMSE and SMAPE values. In this manner, this work provides a summary of prediction techniques, including the type of chaotic time series, predicted steps ahead, and the prediction error. Among those techniques, the echo state network (ESN), long short-term memory, artificial neural network and convolutional neural network are compared with similar conditions to predict up to ten steps ahead of Lorenz-chaotic time series. The comparison among these prediction techniques include RMSE and SMAPE values, training and testing times, and required memory in each case. Finally, considering RMSE and SMAPE, with relatively few neurons in the reservoir, the performance comparison shows that an ESN is a good technique to predict five to fifteen steps ahead using thirty neurons and taking the lowest time for the tracking and testing cases.

KEYWORDS
Chaotic time series
Neural network
Echo state network
Long short-term memory
RMSE
Prediction technique

INTRODUCTION

Chaos has been a research area that includes several physical phenomena that can be modeled by deterministic mathematical equations, applied to real life problems and predicted applying artificial intelligence based techniques. For example: one can take a chaotic system as the well-known Lorenz oscillator to generate chaotic time series; afterwards, one can use the time series to try to predict several steps ahead applying prediction techniques. In this prediction problem one has the challenge of choosing the appropriate technique, which depends on the nature of the data to validate the prediction, e.g. some data can have slow variations and others fast changes in their dynamics. Some examples of chaotic time series in the real world are for example: sunspots, water run-off, electric changes, temperature, rainfalls, voice signals Lau and Wu (2008); Yang *et al.* (2005); Dhanya and Nagesh Kumar (2010); Jingjing *et al.* (2018), and so on. Clearly, these data is different and therefore the challenge is the development or application of known prediction techniques that guarantee a large prediction horizon with minimum error.

Some of the main characteristics that try to quantify chaotic behavior was introduced by Li and Yorke (1975). From this seminal work, one understand important concepts as fractal dimension, Lyapunov exponents, Fourier transform, Hilbert transform and the reconstruction of an attractor Liu (2010). Another seminal work was introduced by Wolf *et al.* (1985), for determining Lyapunov exponents from a time series, where the chaotic time series can be experimental or taken from simulation. In this manner, one can evaluate the Lyapunov exponent of a chaotic time series to validate if it is chaotic or not, and therefore, it is chaotic if the Lyapunov exponent is positive. On another point of view, it is said that chaotic time series present characteristics seemingly unpredictable due to their complexity Han *et al.* (2019c), and due to their high sensitivity to the initial conditions, as shown by Wolf *et al.* (1985).

Manuscript received: 13 May 2022,
Revised: 18 July 2022,
Accepted: 22 July 2022.

[1] josuealexis15@hotmail.com
[2] erechy@uv.mx
[3] amgonzalezz@uqvirtual.edu.co
[4] etlelo@inaoep.mx (**Corresponding Author**)

One can find a huge number of chaotic time series from physical phenomena or generated from mathematical models. For instance, the authors in Liu (2010) talk about tornadoes and human brain, in which the challenge is predicting the future behavior, thus requiring the development of prediction techniques. Fortunately, nowadays one can find contributions to chaotic time series prediction applying artificial intelligence, statistics, mathematics, electronics among other research areas. On this direction, some authors have shown the usefulness of applying Artificial Neural Networks (ANN) Ong and Zainuddin (2019); Chen and Han (2013); Pano-Azucena *et al.* (2021), fuzzy logic Miranian and Abdollahzade (2013); Heydari *et al.* (2016); Goudarzi *et al.* (2016), Bayes theoremLi *et al.* (2016); Wang *et al.* (2020b), Machine Learning (ML) Alemu (2018); Gromov and Borisenko (2015), multilayer Perceptrons Dalia Pano-Azucena *et al.* (2018); Zhao *et al.* (2014), recurrent neural networks (RNN) Li *et al.* (2012); Ardalani-Farsa and Zolfaghari (2010); Chandra and Zhang (2012); Xu *et al.* (2019), linear and nonlinear filters Wu and Song (2013); Ma *et al.* (2017); Yumei *et al.* (2019), optimization by evolutionary computation Samanta (2011); Chandra *et al.* (2017); Guo *et al.* (2016b), approximation by recursion Wang *et al.* (2017); Li-yun (2010); Han *et al.* (2019b), statistical methods Kurogi *et al.* (2018); Xu *et al.* (2019); Jokar *et al.* (2019), Wavelet transform Zhongda *et al.* (2017); Feng *et al.* (2019b), Lyapunov exponents Yong (2013), computer algorithms Guo *et al.* (2020); Hua *et al.* (2013); Jingjing *et al.* (2018); Zhou *et al.* (2017) and hybrid architectures Xiao *et al.* (2019); Fu *et al.* (2010); Han *et al.* (2017).

Among these techniques, the optimization by evolutionary computation and hybrid architectures have shown good results. In the case of optimization by evolutionary computation, one can find the application of Particle Swarm Optimization (PSO) Eberhart and Kennedy (1995), Differential Evolution (DE) Price *et al.* (2006), Cuckoo Search Yang and Deb (2010), Ant Colony Optimization (ACO) Dorigo *et al.* (2006), Fruit Fly Optimization Algorithm Xing and Gao (2014), Whale Optimization Algorithm Mirjalili and Lewis (2016), grey wolf optimizer Mirjalili *et al.* (2014) and co-evolution, where different optimization methods work together.

In the case of hybrid architectures for chaotic time series prediction, the most known are: Bayes theorem Swinburne (2004), Echo State Network (ESN) Jaeger (2007), ANN Drew and Monson (2000), Wavelet transform Zhang (2019a), long short-term memory (LSTM) and Least Square Support Vector Machine (LSSVM) Suykens and Vandewalle (1999).

In this manner, this paper provides a summary on chaotic time series prediction techniques and compares the performance of four techniques based on neural networks to predict chaotic time series from Lorenz chaotic system. The next section shows the most used models of Lorenz system and Mackey-Glass, and others, and shows a Table summarizing different prediction techniques, comparing the predicted steps, data used for the prediction and the associated root-mean-square error (RMSE) for each case. Afterwards, this paper compares four prediction techniques based on neural networks, namely: ESN, LSTM, ANN and 1-Dimension Convolutional Neural Network (1D-CNN). The prediction results are shown in the section before concluding this work.

TECHNIQUES FOR CHAOTIC TIME SERIES PREDICTION

In the current state of the art, one can find different techniques oriented to predict chaotic time series. The following papers were used for the classification of prediction techniques, predicted steps, number of points used for the prediction, and the associated RMSE: Alemu (2018); Shinozaki *et al.* (2020); Zhang and Jiang (2020); Su and Yang (2021); Zhang *et al.* (2020). The chaotic time series data was mainly taken from two chaotic systems: *Lorenz* and *Mackey-Glass*.

1. **Lorenz**:
   This is a deterministic system modeled by three ordinary differential equations (ODEs) introduced by Lorenz (1963), and given by (1), where chaotic behavior exists by setting $\sigma = 10$, $\rho = 28$ and $\beta = \frac{8}{3}$.

$$\begin{aligned} \frac{dx(t)}{dt} &= \sigma[y(t) - x(t)] \\ \frac{dy(t)}{dt} &= x(t)[\rho - z(t)] - y(t) \\ \frac{dz(t)}{dt} &= x(t)y(t) - \beta z(t) \end{aligned} \tag{1}$$

2. **Mackey-Glass**:
   This chaotic system was introduced by Mackey and Glass (1977), and denoted by (2), where τ is a delay parameter, and it can be set to $\tau \leq 4.43$ to produce a fixed point, $4.43 \leq \tau \leq 13.3$ to produce a stable limit cycle, $13.3 \leq \tau \leq 16.8$ to produce a double limit attraction, and $16.8 \leq \tau$ to generate chaotic behavior.

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1 + x^c(t-\tau)} - bx(t) \tag{2}$$

When simulating a chaotic system, the amplitudes of the state variables can be as large as possible, however, for hardware implementation, it is desired to have amplitudes within the range $[-1, 1]$ or $[0, 1]$. In the validation of the steps predicted by each technique, the authors use different errors, such as: RMSE, Mean Square Error (MSE), Mean Absolute Error (MAE), Normalized RMSE (NRMSE), R^2, among others. However, in the majority of works, the most used measure is RMSE, which is defined by (3), where N is the total of attributes, $\tilde{y}_n$ is the predicted value and y_n^{target} the reference value.

$$RMSE = \sqrt{\frac{\Sigma_{n=1}^{N} \left(\tilde{y}_n - y_n^{target}\right)^2}{N}} \tag{3}$$

In addition, Symmetric Mean Absolute Percentage Error (SMAPE) is implemented like an accuracy measure based on percentage errors. This error is described in equation (4) and indicates the percent of accuracy of the real value versus the forecast value in descendent form, where N is the total of attributes, F_n is the predicted value and A_n is the actual value.

$$SMAPE = \frac{100\%}{n} \Sigma_{n=1}^{N} \frac{|F_n - A_n|}{\frac{(|F_n| + |A_n|)}{2}} \tag{4}$$

In Table 1, we list some prediction techniques including the type of chaotic data used by the associated technique, the predicted steps ahead, number of test points, and RMSE. It can be appreciated that hybrid and optimized techniques have low RMSE, and also, the low errors are associated to the techniques predicting 1 step ahead of chaotic time series. The minimum number of points for testing each technique is 500.

Table 1 Prediction techniques for chaotic time series, ordered from the lowest to the highest RMSE value.

Technique	Chaotic serie	Prediction	Test data	RMSE
Combining the phase space reconstruction and fuzzy logic Gholizade-Narm and Shafiee-Chafi (2015)	Mackey-Glass	1 step	600	2.26E-10
Hybrid Empirical Mode Decomposition - Neural Networks (HEMD-NN) Tang *et al.* (2020)	Mackey-Glass	1 step	2,000	5.31E-08
Efficient Extreme Learning Machine - Differential Evolution (EELM-DE) Guo *et al.* (2016b)	Lorenz	1 step	500	7.67E-08
Kernel Local Polynomial coefficient autoregressive Prediction (KLPP) Su and Li (2015b)	Henon	1 step	500	8.44E-07
Hybrid Elman-NARX neural networks Ardalani-Farsa and Zolfaghari (2010)	Mackey-Glass	1 step	1,000	3.72E-05
Radial Basis Function (RBF) neural network Zhang *et al.* (2013)	Drift sensor	2 step	4,000	4.87E-05
ESN optimized by Selective Opposition Grey Wolf Optimizer (SOGWO-ESN) Chen and Wei (2021)	Mackey-Glass	25 step	800	1.46E-04
Artificial Neural Networks (ANNs), Adaptive Neuro-Fuzzy Inference System (ANFIS) and Least-Squares Support Vector Machines (LSSVM) Dalia Pano-Azucena *et al.* (2018)	Chaotic system	6 step	2,000	2.98E-04
Local Neuro-Fuzzy (LNF) - Least-Squares Support Vector Machines (LSSVMs) Miranian and Abdollahzade (2013)	Mackey-Glass	6 step	500	7.90E-04
Local Functional Coefficient Autoregressive (LFAR) Su and Li (2015a)	Mackey-Glass	500 step	500	1.30E-03
Structured Manifold - Broad Learning System (SM-BLS) Han *et al.* (2019a)	Lorenz	10 step	$\approx$ 4,400	2.45E-03
Hierarchical Delay-Memory Echo State Network (HDESN) Na *et al.* (2021)	Lorenz	12 step	2,000	2.65E-03
The Elman recurrent networkChandra and Zhang (2012)	Mackey-Glass	500 step	500	6.33E-03
Local Volterra model based on phase points clustering Han *et al.* (2018)	Lorenz	50 step	4,976	8.10E-03

PREDICTION TECHNIQUES BASED ON NEURAL NETWORKS

This section shows a comparison among prediction techniques based on most used neural networks.

Echo State Network

The prediction technique based on ESN was introduced by Jaeger (2007). It becomes to behave as a recurrent neural network and includes a reservoir that assigns random weights while a certain percentage or neurons are connected by accomplishing the property of echo, as shown by Lukoševičius (2012). The update equations are given in equations (5) and (6). In these equations $x(n)$ denotes the activation vector of the neurons in the reservoir, where n is the value of each neuron in the reservoir, α is the leaking rate denoted by $\in (0, 1]$ for the training. $\tilde{x}(n)$ is the update for each n, where $tanh(\cdot)$ holds the vertical concatenation of matrix W^{in} and W represents the inputs and recurrent weights of the matrices. The output layer is defined by equations (7) and (8).

$$\tilde{x}(n) = tanh(\mathrm{W}^{in}[1; u(n)] + \mathrm{W} \times (n-1)) \tag{5}$$

$$x(n) = (1-\alpha) \times (n-1) + \alpha\tilde{x}(n) \tag{6}$$

$$W^{out} = Y^{target} X^T (XX^T + \beta I)^{-1} \tag{7}$$

$$y(n) = W^{out}[1; u(n); x(n)] \tag{8}$$

In equation (8), $y(n)$ is the output layer, W^{out} is the weights output matrix determined by *Ridge regression* or also known as *Tikhonov* regularization, where β it's regularization coefficient. On other hand, $[\cdot;\cdot;\cdot]$ holds the verticality in the concatenation of the vector as mentioned above; X is the collect data of W (it mean the percent of connection in the reservoir).The simulation of this prediction technique includes a reservoir of 30 neurons and a spectral radius (SP) of 2.5.

Long Short-Term Memory

The technique known as Long Short-Term Memory (LSTM) is a kind of recurrent neural network that was introduced by Hochreiter and Schmidhuber (1997). Its main characteristic is the ability to retain one state of a sequence in a long term. An LSTM has three inputs and two outputs: x_t is the current input value as denoted by equations (10) and (11); while at the same time shares the input with h_{t-1}, that is the previous output value of the net, as described by equations (10) and (11). c_{t-1} denotes the input to the *cell state*; the outputs h_t are denoted by equations (13) and (14), and C_t is the unitary state of the current LSTM net given by equation (12). This LSTM in addition includes update gates of information to forget and update the cell state values. The description of the *forget gate* is given in f_t by equation (9) and the *update gate* is given in C_t by (12).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{9}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{10}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{11}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{12}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{13}$$

$$h_t = o_t * tanh(C_t) \tag{14}$$

In these equations σ is a sigmoid function scaled within the values $[0, 1]$ for the updating of the C_t (*cell state*) for the consumption of the next time step LSTM. *tanh* denotes the activation function, $W_{[\cdot]}$ is the weight matrix for learning, $b_{[\cdot]}$ is the bias or every neuronal network in the LSTM, and x_t denotes the inputs to the LSTM net, h_{t-1} and c_{t-1} are the inputs from the previous time step and f_t means the forget gate. The simulation of this technique consists of the series connection of four LSTM.

Artificial Neural Network

The ANN was introduced by McCulloch and Pitts (1943), as a mathematical model described by a bio-inspiration of the neurons in the human brain. An ANN consists of an array of artificial neurons connected in a *feed forward* way. In this manner, it consists of at least three layers, namely: input layer, hidden layer and output layer. The input layer can be described by vector x_i; in the hidden layer take place the operations evaluated by the weights w_i and bias b, it includes the activation functions to each neuron denoted as f. The hidden layer operates on equation (15), and it can include more than one layer. In the output layer, the last evaluations take place to provide the learned data. The training of an ANN consists of epochs, and the most used training method is known as *Backpropagation* that is denoted by equations (16) and (17).

$$\sum_{i=1}^{i=n} (w_i * x_i) + b \tag{15}$$

$$E = \frac{\sum_{i=1}^{i=n} (t_i - a_i)^2}{2} \tag{16}$$

$$\Delta \mathrm{W} = -\alpha \frac{\partial EW}{\partial W} \tag{17}$$

In this case, equation (16) evaluates the mean square error of the target t_i and the output of the neuron a_i, which updates the weights by (17) when the net is back-propagated to learn in each epoch. The simulation of this technique was performed considering a hidden layer of 20, 15 and 10 neurons.

Convolutional Neural Network

The Convolutional Neural Network (CNN) is a kind of ANN introduced by Fukushima (1980). The difference with an ANN is the application devoted to bidirectional matrices, being quite effective for artificial vision tasks. However, its application is also suitable for time series prediction, plain images and signals from functional magnetic resonance images. The CNN consists of the main layers: Convolutional layer, which performs the convolution of the inputs with a kernel given in equation (18); the Max-pooling layer, which extracts the main characteristics form the convolution; and the third layer is a fully connected network (feed forward).

$$Y_j = g(b_j + \sum_i K_{ij} \otimes Y_i) \tag{18}$$

In equation (18), Y_j is the output of neuron j evaluated through a linear combination of the outputs Y_i of the neurons in the previous layer, each one operated with the convolutional core K_{ij} corresponding to that connection. This value is added to b_j and afterwards send to an activation function $g(\cdot)$ of non-linear type. For chaotic time series prediction, the CNN has a kernel that moves in one direction, i.e. guided by the time series. The simulation of this technique was performed using a Max-pooling layer (MP) and 50 neurons that are full-connected among them.

SIMULATION RESULTS

The simulation of the four prediction techniques described in the previous section, was performed using a personal computer with Intel i5-11400H processor of 64 bit at 2.70 GHz, with 8 Gb of RAM. The four techniques have similar characteristics to perform the prediction and was executed each one five times. In this manner, the training was executed using a random seed trying to get similar results. In all the cases, the Lorenz and Mackey-Glass systems was simulated to generate a data of 1500 points that were used for the training and 800 points for the test during the prediction, omitting the first 200 points that are the transitory state of the chaotic system. The four techniques were executed using a leaking rate of 0.001 with 180 epochs for the learning, except for the ESN. The prediction of the steps ahead was performed in an adjacent way with respect to the inputs and predicted steps.

The prediction capabilities of the four techniques is given considering four characteristics: (I) predicted steps, (II) errors (RMSE and SMAPE), (III) training and test time, and (IV) memory required during the training and test, as listed in Tables 2 and 3. In each prediction technique, five runs were executed for each step prediction, reporting the best result of this five executed. The RMSE is the total over the 800 test data of the predicted values. As one can see, the lowest RMSE at one step is provided by LSTM, while the lowest RMSE with the highest predicted steps ahead (15) was provided by CNN. However, the ESN provides the results in general with low RMSE for the prediction with different steps ahead, in addition the SMAPE presents de low variance that others models. Figures 1, 2, 3 and 4 show the better prediction for the chaotic time series results reaching 15 steps ahead applying ESN, LSTM, ANN and CNN techniques, respectively. In the experiments, considering Lorenz time series the techniques reported lower RMSE and SMAPE than when using Mackey-Glass time series, as shown in Tables 2 and 3.

The determination of the maximum predicted steps ahead given in Tables 2 and 3 was done according to the steps ahead (1, 3, 5, 10 and 15 steps). Details of the topologies of each prediction technique are also given in the Tables. For example, one can see the quantity of layers and neurons in each case. It is worth noting the stability of ESN, considering the RMSE and SMAPE, it has low execution time and memory requirement with respect to the results provided by the other prediction techniques. From the results shown in Figures 1, 2, 3, and 4, one can see that some models present high variance in the prediction, as reported in Tables 2 and 3, where SMAPE presents high values. One can also see that the prediction using Lorenz time series is much better providing low RMSE and SMAPE for ESN. However, when using the Mackey-Glass time series the prediction techniques present a similar RMSE result, as shown Table 3. ESN presented a low accuracy in Mackey-Glass with respect to the Lorenz time series. The reason for this are the values of the parameters, since it is a time series with a different behavior. Parameters such as number of neurons, spectral radius, among others, must be adjusted to obtain good results, compared to the other three models, since they adapt to the series with the passage of time. Finally, in Table 4 we show the best results of our experiments with each prediction technique and compared with results in the state of the art.

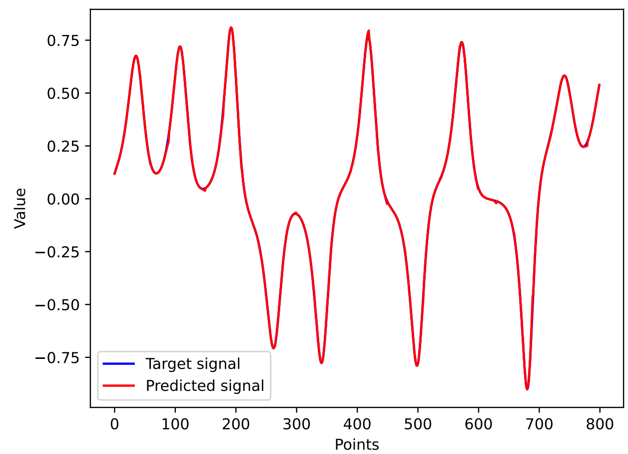


Figure 1 Lorenz time series prediction results by ESN reaching 15 steps ahead.

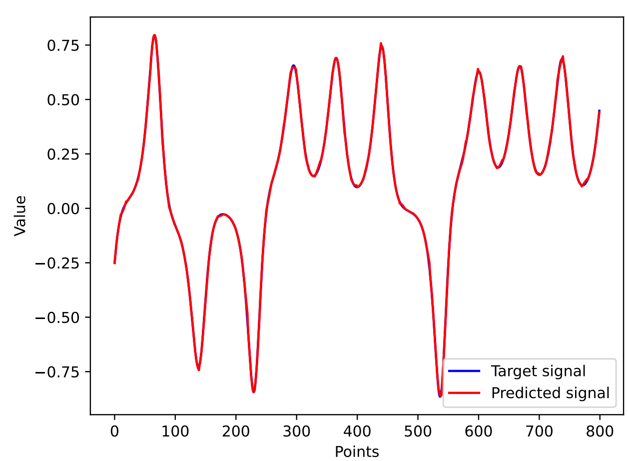


Figure 2 Lorenz time series prediction results by LSTM reaching 15 steps ahead.

Table 2 Comparison of the best executions in the four prediction techniques with Lorenz time series, listing the predicted steps ahead, RMSE, SMAPE, training and testing times, and training and testing memory.

ESN:	**Step ahead**	**Training-time**	**Testing-time**	**Training-memory**	**Testing-memory**	**RMSE**	**SMAPE**
Train=1,500	1	0.0156 Sec.	0.0625 Sec.	1.7728 Mb	0.2901 Mb	2.86E-02	12.52%
Test=800	3	0.0375 Sec.	0.0531 Sec.	1.8654 Mb	0.3746 Mb	2.11E-03	4.69%
Neurons=30	5	0.0317 Sec.	0.2157 Sec.	1.9689 Mb	0.5094 Mb	8.21E-04	0.75%
SP=2.5	10	0.0312 Sec.	0.0562 Sec.	2.1878 Mb	0.9013 Mb	1.30E-03	0.86%
Leaking rate=0.001	15	0.0316 Sec.	0.0467 Sec.	2.3818 Mb	1.2797 Mb	5.76E-03	1.47%
LSTM:	**Step ahead**	**Training-time**	**Testing-time**	**Training-memory**	**Testing-memory**	**RMSE**	**SMAPE**
Train=1,500	1	364.0543 Sec.	0.9687 Sec.	5.2716 Mb	1.9529 Mb	2.22E-02	11.63%
Test=800	3	404.6772 Sec.	0.9218 Sec.	5.5292 Mb	1.9757 Mb	3.04E-02	4.71%
LSTMs=4	5	446.3018 Sec.	0.9278 Sec.	5.3843 Mb	1.9955 Mb	4.51E-03	3.17%
Epochs=180	10	556.7754 Sec.	0.9322 Sec.	5.2363 Mb	2.2375 Mb	6.89E-03	7.64%
Leaking rate=0.001	15	365.8241 Sec.	0.9443 Sec.	5.2806 Mb	1.9483 Mb	1.50E-02	7.86%
ANN:	**Step ahead**	**Training-time**	**Testing-time**	**Training-memory**	**Testing-memory**	**RMSE**	**SMAPE**
Train=1,500	1	265.9227 Sec.	0.1718 Sec.	1.7933 Mb	0.5118 Mb	2.59E-02	14.91%
Test=800	3	266.1119 Sec.	0.1812 Sec.	1.7667 Mb	0.6770 Mb	5.02E-03	11.63%
Layers=20,15,10	5	262.3324 Sec.	0.1624 Sec.	1.7926 Mb	0.7977 Mb	5.85E-03	3.65%
Epochs=180	10	263.9948 Sec.	0.1673 Sec.	1.8077 Mb	1.1445 Mb	6.28E-03	8.65%
Leaking rate=0.001	15	263.7734 Sec.	0.1685 Sec.	1.7948 Mb	1.4987 Mb	7.07E-03	6.81%
CNN:	**Step ahead**	**Training-time**	**Testing-time**	**Training-memory**	**Testing-memory**	**RMSE**	**SMAPE**
Train=1,500	1	268.1515 Sec.	0.1781 Sec.	1.8653 Mb	0.5576 Mb	2.27E-02	12.01%
Test=800	3	265.2860 Sec.	0.1875 Sec.	1.8602 Mb	0.6726 Mb	4.72E-03	6.76%
Layers=1MP,50	5	269.0973 Sec.	0.1866 Sec.	1.7919 Mb	0.8411Mb	6.64E-03	6.72%
Epochs=180	10	274.2548 Sec.	0.1875 Sec.	1.8689 Mb	1.1925 Mb	7.41E-03	8.92%
Leaking rate=0.001	15	274.9681 Sec.	0.1866 Sec.	1.8156 Mb	1.5433 Mb	5.45E-03	6.32%

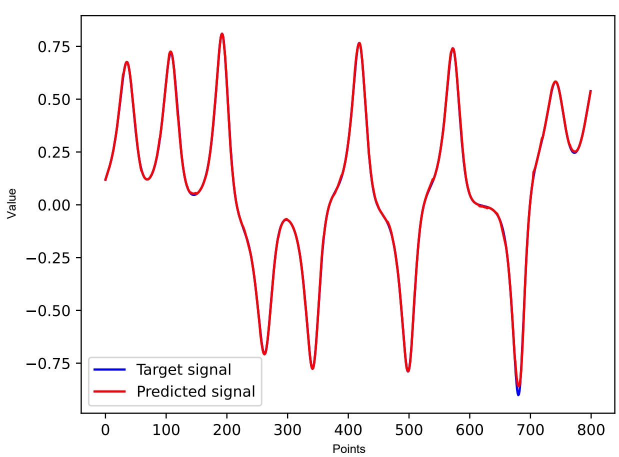


Figure 3 Lorenz time series prediction results by ANN reaching 15 steps ahead.

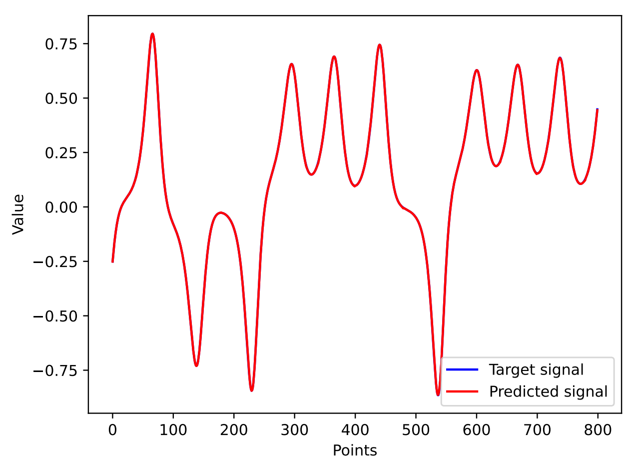


Figure 4 Lorenz time series prediction results by CNN reaching 15 steps ahead.

Table 3 Comparison of the best executions in the four prediction techniques with Mackey-Glass time series, listing the predicted steps ahead, RMSE, SMAPE, training and testing times, and training and testing memory.

ESN:	Step ahead	Training-time	Testing-time	Training-memory	Testing-memory	RMSE	SMAPE
Train=1,500	1	0.2656 Sec.	0.3125 Sec.	2.9410 Mb	0.3280 Mb	3.74E-02	8.23%
Test=800	3	0.0468 Sec.	0.0624 Sec.	3.0219 Mb	0.3865 Mb	1.49E-02	2.76%
Neurons=30	5	0.0312 Sec.	0.0468 Sec.	3.1151 Mb	0.5333 Mb	1.96E-02	3.14%
SP=2.5	10	0.0312 Sec.	0.0624 Sec.	3.3470 Mb	0.9001 Mb	8.47E-02	12.01%
Leaking rate=0.001	15	0.0312 Sec.	0.0624 Sec.	2.5823 Mb	1.2691 Mb	7.79E-02	11.41%
LSTM:	**Step ahead**	**Training-time**	**Testing-time**	**Training-memory**	**Testing-memory**	**RMSE**	**SMAPE**
Train=1,500	1	363.7754 Sec.	0.9218 Sec.	5.2639 Mb	1.9470 Mb	3.49E-02	7.46%
Test=800	3	412.3002 Sec.	0.9446 Sec.	5.1749 Mb	1.9738 Mb	1.27E-02	2.26%
LSTMs=4	5	462.1199 Sec.	1.2499 Sec.	7.9011 Mb	2.4232 Mb	2.52E-02	3.96%
Epochs=180	10	555.2916 Sec.	0.9218 Sec.	5.1964 Mb	2.2387 Mb	2.38E-02	4.72%
Leaking rate=0.001	15	647.9304 Sec.	1.2812 Sec.	5.1711 Mb	2.6389 Mb	1.82E-02	3.60%
ANN:	**Step ahead**	**Training-time**	**Testing-time**	**Training-memory**	**Testing-memory**	**RMSE**	**SMAPE**
Train=1,500	1	266.2632 Sec.	0.1562 Sec.	1.7642 Mb	0.5119 Mb	3.59E-02	7.86%
Test=800	3	263.4166 Sec.	0.1562 Sec.	1.7559 Mb	0.6515 Mb	1.58E-02	2.37%
Layers=20,15,10	5	268.1604 Sec.	0.1874 Sec.	1.7469 Mb	0.7927 Mb	3.30E-02	4.65%
Epochs=180	10	266.1194 Sec.	0.1562 Sec.	1.7969 Mb	1.1753 Mb	1.52E-02	2.58%
Leaking rate=0.001	15	268.1438 Sec.	0.1528 Sec.	2.0190 Mb	0.1528 Mb	1.04E-02	2.00%
CNN:	**Step ahead**	**Training-time**	**Testing-time**	**Training-memory**	**Testing-memory**	**RMSE**	**SMAPE**
Train=1,500	1	267.5832 Sec.	0.1875 Sec.	1.8019 Mb	0.5631 Mb	3.51E-02	8.45%
Test=800	3	264.8904 Sec.	0.1875 Sec.	1.8881 Mb	0.7009 Mb	2.14E-02	4.37%
Layers=1MP,50	5	269.0176 Sec.	0.2031 Sec.	1.7811 Mb	0.8507 Mb	3.63E-02	6.83%
Epochs=180	10	272.9374 Sec.	0.1718 Sec.	1.7973 Mb	1.1921 Mb	1.61E-02	2.89%
Leaking rate=0.001	15	272.2324 Sec.	0.1875 Sec.	1.7894 Mb	1.5429 Mb	9.52E-03	2.14%

Table 4 Comparison of our better results with the state of the art

Technique	Chaotic serie	Prediction	RMSE
Our approach with LSTM	Lorenz	1 step	2.22E-02
Deep Hybrid Neural Network with Differential Neuroevolution Huang *et al.* (2020)	Lorenz	1 step	7.56E-02
Our approach with ESN	Lorenz	5 steps	8.21E-04
Adaptive Sparse Quantization Kernel Least Mean Square Algorithm Zhao *et al.* (2021)	Beijing PM 2.5	5 steps	3.15E-02
Improved Kernel Recursive Least Squares Algorithm Han *et al.* (2019b)	Lorenz	5 steps	4.41E-02
Co-evolutionary predictive algorithm Chandra *et al.* (2017)	Mackey-Glass	5 steps	5.90E-02
Our approach with ESN	Lorenz	10 steps	1.30E-03
Structured Manifold - Broad Learning System (SM-BLS) Han *et al.* (2019a)	Lorenz	10 steps	2.45E-03
Robust manifold broad learning system for large-scale noisy chaotic time series prediction Feng *et al.* (2019a)	Lorenz	10 steps	1.82E-01

CONCLUSION

This paper showed the state of the art in chaotic time series prediction using different prediction techniques. From Table 1, it was observed the usefulness of neural networks, so that four techniques were chosen to perform the prediction of time series taken data from Lorenz and Mackey-Glass systems. Tables 2 and 3 summarizes the prediction results provided by applying four techniques that are based on ESN, LSTM, ANN and CNN. As a result, one can see that the ESN is the technique providing better prediction results in its stability of results in the five executions realized. In addition, ESN obtained the low RMSE and SMAPE values. This means that the results provided by ESN have the lower variance in average compared to the other prediction technqiues, and it also requires lower computing resources.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Availability of data and material

Not applicable.

LITERATURE CITED

Alemu, M. N., 2018 A fuzzy model for chaotic time series prediction. International Journal of Innovative Computing Information and Control **14**: 1767–1786.

Ardalani-Farsa, M. and S. Zolfaghari, 2010 Chaotic time series prediction with residual analysis method using hybrid elman-narx neural networks. Neurocomputing **73**: 2540–2553.

Chandra, R., Y.-S. Ong, and C.-K. Goh, 2017 Co-evolutionary multi-task learning with predictive recurrence for multi-step chaotic time series prediction. Neurocomputing **243**: 21–34.

Chandra, R. and M. Zhang, 2012 Cooperative coevolution of elman recurrent neural networks for chaotic time series prediction. Neurocomputing **86**: 116–123.

Chen, D. and W. Han, 2013 Prediction of multivariate chaotic time series via radial basis function neural network. Complexity **18**: 55–66.

Chen, H.-C. and D.-Q. Wei, 2021 Chaotic time series prediction using echo state network based on selective opposition grey wolf optimizer. Nonlinear Dynamics **104**: 3925–3935.

Cheng, W., Y. Wang, Z. Peng, X. Ren, Y. Shuai, *et al.*, 2021 High-efficiency chaotic time series prediction based on time convolution neural network. Chaos Solitons & Fractals **152**.

Dalia Pano-Azucena, A., E. Tlelo-Cuautle, S. X. D. Tan, B. Ovilla-Martinez, and L. Gerardo de la Fraga, 2018 Fpga-based implementation of a multilayer perceptron suitable for chaotic time series prediction. Technologies **6**.

Dhanya, C. and D. Nagesh Kumar, 2010 Nonlinear ensemble prediction of chaotic daily rainfall. Advances in Water Resources **33**: 327–347.

Dorigo, M., M. Birattari, and T. Stutzle, 2006 Ant colony optimization. IEEE computational intelligence magazine **1**: 28–39.

Drew, P. J. and J. R. Monson, 2000 Artificial neural networks. Surgery **127**: 3–11.

Eberhart, R. and J. Kennedy, 1995 Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks*, volume 4, pp. 1942–1948, Citeseer.

Feng, S., W. Ren, M. Han, and Y. W. Chen, 2019a Robust manifold broad learning system for large-scale noisy chaotic time series prediction: A perturbation perspective. Neural Networks **117**: 179–190.

Feng, T., S. Yang, and F. Han, 2019b Chaotic time series prediction using wavelet transform and multi-model hybrid method. Journal of Vibroengineering **21**: 1983–1999.

Fu, Y.-Y., C.-J. Wu, J.-T. Jeng, and C.-N. Ko, 2010 Arfnns with svr for prediction of chaotic time series with outliers. Expert Systems with Applications **37**: 4441–4451.

Fukushima, K., 1980 Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics **36**: 193–202.

Ganjefar, S. and M. Tofighi, 2018 Optimization of quantum-inspired neural network using memetic algorithm for function approximation and chaotic time series prediction. Neurocomputing **291**: 175–186.

Gholizade-Narm, H. and M. R. Shafiee-Chafi, 2015 Using repetitive fuzzy method for chaotic time series prediction. Journal Of Intellıgent & Fuzzy Systems **28**: 1937–1946.

Goudarzi, S., M. B. Khodabakhshi, and M. H. Moradi, 2016 Interactively recurrent fuzzy functions with multi objective learning and its application to chaotic time series prediction. Journal Of Intellıgent & Fuzzy Systems **30**: 1157–1168.

Gromov, V. A. and E. A. Borisenko, 2015 Predictive clustering on non-successive observations for multi-step ahead chaotic time series prediction. Neural Computing & Applications **26**: 1827–1838.

Gromov, V. A. and A. N. Shulga, 2012 Chaotic time series prediction with employment of ant colony optimization. Expert Systems With Applıcations **39**: 8474–8478.

Guo, F., L. Lin, and C. Wang, 2016a Novel continuous function prediction model using an improved takagi-sugeno fuzzy rule and its application based on chaotic time series. Engıneering Applications Of Artificial Intelligence **55**: 155–164.

Guo, W., T. Xu, and Z. Lu, 2016b An integrated chaotic time series prediction model based on efficient extreme learning machine and differential evolution. Neural Computing & Applications **27**: 883–898.

Guo, X., Y. Sun, and J. Ren, 2020 Low dimensional mid-term chaotic time series prediction by delay parameterized method. Information Sciences **516**: 1–19.

Han, F., S. Yang, and S. Song, 2018 Local volterra multivariable chaotic time series multi-step prediction based on phase points clustering. Journal of Vibroengineering **20**: 2486–2503.

Han, M., S. Feng, C. L. P. Chen, M. Xu, and T. Qiu, 2019a Structured manifold broad learning system: A manifold perspective for large-scale chaotic time series analysis and prediction. IEEE Transactıons On Knowledge And Data Engıneerıng **31**: 1809–1821.

Han, M., W. Li, S. Feng, T. Qiu, and C. L. P. Chen, 2021 Maximum information exploitation using broad learning system for large-scale chaotic time-series prediction. IEEE Transactions On Neural Networks And Learning Systems **32**: 2320–2329.

Han, M., R. Zhang, and M. Xu, 2017 Multivariate chaotic time series prediction based on elm-plsr and hybrid variable selection algorithm. Neural Processing Letters **46**: 705–717.

Han, M., S. Zhang, M. Xu, T. Qiu, and N. Wang, 2019b Multivariate chaotic time series online prediction based on improved kernel recursive least squares algorithm. IEEE Transactions On Cybernetics **49**: 1160–1172.

Han, M., K. Zhong, T. Qiu, and B. Han, 2019c Interval type-2 fuzzy neural networks for chaotic time series prediction: A concise overview. IEEE Transactions on Cybernetics **49**: 2720–2731.

Heydari, G., M. Vali, and A. A. Gharaveisi, 2016 Chaotic time series prediction via artificial neural square fuzzy inference system.

Expert Systems with Applications **55**: 461–468.
Hochreiter, S. and J. Schmidhuber, 1997 Long Short-Term Memory. Neural Computation **9**: 1735–1780.
Hua, Q., M. Wen-Tao, Z. Ji-Hong, and C. Ba-Dong, 2013 Kernel least mean kurtosis based online chaotic time series prediction. Chinese Physics Letters **30**.
Huang, W., Y. Li, and Y. Huang, 2020 Deep hybrid neural network and improved differential neuroevolution for chaotic time series prediction. IEEE Access **8**: 159552–159565.
Jaeger, H., 2007 Echo state network. scholarpedia **2**: 2330.
Jian-Ling, Q., W. Xiao-Fei, Q. Yu-Chuan, G. Feng, and D. Ya-Zhou, 2014 An improved local weighted linear prediction model for chaotic time series. Chinese Physics Letters **31**.
Jianshan, L., W. Changming, Z. Aijun, and X. Xiaomin, 2012 Residual gm(1,1) model-based prediction method for chaotic time series. Journal of Grey System **24**: 379–388.
Jingjing, L., Z. Qijin, Z. Yumei, W. Xiaojun, W. Xiaoming, *et al.*, 2018 Hidden phase space reconstruction: A novel chaotic time series prediction method for speech signals. Chinese Journal of Electronics **27**: 1221–1228.
Jokar, M., H. Salarieh, and A. Alasty, 2019 On the existence of proper stochastic markov models for statistical reconstruction and prediction of chaotic time series. Chaos Solitons & Fractals **123**: 373–382.
Kurogi, S., M. Toidani, R. Shigematsu, and K. Matsuo, 2018 Performance improvement via bagging in probabilistic prediction of chaotic time series using similarity of attractors and loocv predictable horizon. Neural Computing & Applications **29**: 341–349.
Lau, K. W. and Q. H. Wu, 2008 Local prediction of non-linear time series using support vector regression. Pattern Recogn. **41**: 1539–1547.
Li, D., M. Han, and J. Wang, 2012 Chaotic time series prediction based on a novel robust echo state network. IEEE Transactions On Neural Networks and Learning Systems **23**: 787–799.
Li, Q. and R.-C. Lin, 2016 A new approach for chaotic time series prediction using recurrent neural network. Mathematical Problems in Engineering **2016**.
Li, T.-Y. and J. A. Yorke, 1975 Period three implies chaos. The American Mathematical Monthly **82**: 985–992.
Li, Y., X. Jiang, H. Zhu, X. He, S. Peeta, *et al.*, 2016 Multiple measures-based chaotic time series for traffic flow prediction based on bayesian theory. Nonlinear Dynamics **85**: 179–194.
Li-yun, S., 2010 Prediction of multivariate chaotic time series with local polynomial fitting. Computers & Mathematiıcs with Applications **59**: 737–744.
Liu, Z., 2010 Chaotic time series analysis. Mathematical Problems in Engineering **2010**: 720190.
Lorenz, E. N., 1963 Deterministic nonperiodic flow. Journal of Atmospheric Sciences **20**: 130 – 141.
Lukoševičius, M., 2012 A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade: Second Edition*, edited by G. Montavon, G. B. Orr, and K.-R. Müller, pp. 659–686, Springer Berlin Heidelberg, Berlin, Heidelberg.
Lv, M., X. Zhang, H. Chen, C. Ling, and J. Li, 2020 An accurate online prediction model for kiln head temperature chaotic time series. IEEE Access **8**: 44288–44299.
Ma, W., J. Duan, W. Man, H. Zhao, and B. Chen, 2017 Robust kernel adaptive filters based on mean p-power error for noisy chaotic time series prediction. Engıneerıng Applıcatıons Of Artıfıcıal Intellıgence **58**: 101–110.
Mackey, M. C. and L. Glass, 1977 Oscillation and chaos in physiological control systems. Science **197**: 287–289.
McCulloch, W. S. and W. Pitts, 1943 A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics **5**: 115–133.
Miranian, A. and M. Abdollahzade, 2013 Developing a local least-squares support vector machines-based neuro-fuzzy model for nonlinear and chaotic time series prediction. IEEE Transactions on Neural Networks and Learning Systems **24**: 207–218.
Mirjalili, S. and A. Lewis, 2016 The whale optimization algorithm. Advances in engineering software **95**: 51–67.
Mirjalili, S., S. M. Mirjalili, and A. Lewis, 2014 Grey wolf optimizer. Advances in engineering software **69**: 46–61.
Na, X., W. Ren, and X. Xu, 2021 Hierarchical delay-memory echo state network: A model designed for multi-step chaotic time series prediction. Engıneerıng Applıcatıons Of Artıfıcıal Intellıgence **102**.
Nguyen, H. M., G. Kalra, T. Jun, and D. Kim, 2020 Chaotic time series prediction using a novel echo state network model with input reconstruction, bayesian ridge regression and independent component analysis. Internatıonal Journal Of Pattern Recognıtıon And Artıfıcıal Intellıgence **34**.
Ong, P. and Z. Zainuddin, 2019 Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series prediction. Applied Soft Computing **80**: 374–386.
Pano-Azucena, A. D., E. Tlelo-Cuautle, B. Ovilla-Martinez, L. G. de la Fraga, and R. Li, 2021 Pipeline fpga-based implementations of anns for the prediction of up to 600-steps-ahead of chaotic time series. Journal Of Cırcuıts Systems And Computers **30**.
Price, K., R. M. Storn, and J. A. Lampinen, 2006 *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media.
Samanta, B., 2011 Prediction of chaotic time series using computational intelligence. Expert Systems With Applications **38**: 11406–11411.
Shi, Y., X. Liu, T. Li, X. Peng, W. Chen, *et al.*, 2010 Chaotic time series prediction using immune optimization theory. International Journal Of Computational Intelligence Systems **3**: 43–60.
Shinozaki, A., T. Miyano, and Y. Horio, 2020 Chaotic time series prediction by noisy echo state network. IEICE Nonlinear Theory and its Applications **11**: 466–479.
Shoaib, B., I. M. Qureshi, Shafqatullah, and Ihsanulhaq, 2014 Adaptive step-size modified fractional least mean square algorithm for chaotic time series prediction. Chinese Physics B **23**.
Su, L. and C. Li, 2015a Local functional coefficient autoregressive model for multistep prediction of chaotic time series. Discrete Dynamics In Nature and Society **2015**.
Su, L. and C. Li, 2015b Local prediction of chaotic time series based on polynomial coefficient autoregressive model. Mathematical Problems in Engineering **2015**.
Su, L. and F. Yang, 2021 Prediction of chaotic time series based on ben-aga model. Complexity **2021**.
Suykens, J. A. and J. Vandewalle, 1999 Least squares support vector machine classifiers. Neural processing letters **9**: 293–300.
Swinburne, R., 2004 Bayes' theorem. Revue Philosophique de la France Et de l **194**.
Tang, L.-H., Y.-L. Bai, J. Yang, and Y.-N. Lu, 2020 A hybrid prediction method based on empirical mode decomposition and multiple model fusion for chaotic time series. Chaos Solitons & Fractals **141**.
Wang, C., H. Zhang, W. Fan, and P. Ma, 2017 A new chaotic time series hybrid prediction method of wind power based on eemd-se and full-parameters continued fraction. Energy **138**: 977–990.

Wang, H. and J. Lian, 2011 Fuzzy prediction of chaotic time series based on fuzzy clustering. Asian Journal Of Control **13**: 576–581.

Wang, H., F. Sun, Y. Cai, and Z. Zhao, 2010 Online chaotic time series prediction using unbiased composite kernel machine via cholesky factorization. Soft Computing **14**: 931–944.

Wang, R., C. Peng, J. Gao, Z. Gao, and H. Jiang, 2020a A dilated convolution network-based lstm model for multi-step prediction of chaotic time series. Computational & Applied Mathematics **39**.

Wang, Y., Z. Man, and M. Lu, 2020b Prediction of energy-efficient production of coalbed methane based on chaotic time series and bayes-least squares-support vector machine. International Journal Of Heat And Technology **38**: 933–940.

Wolf, A., J. B. Swift, H. L. Swinney, and J. A. Vastano, 1985 Determining lyapunov exponents from a time series. Physica D: nonlinear phenomena **16**: 285–317.

Wu, X. and Z. Song, 2013 Multi-step prediction of chaotic time-series with intermittent failures based on the generalized nonlinear filtering methods. Applied Mathematics And Computation **219**: 8584–8594.

Xiao, Y., X. Xie, Q. Li, and T. Li, 2019 Nonlinear dynamics model for social popularity prediction based on multivariate chaotic time series. Physica A-Statistical Mechanics And Its Applications **525**: 1259–1275.

Xin, B. and W. Peng, 2020 Prediction for chaotic time series-based ae-cnn and transfer learning. Complexity **2020**.

Xing, B. and W.-J. Gao, 2014 Fruit fly optimization algorithm. In *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*, pp. 167–170, Springer.

Xu, M., M. Han, T. Qiu, and H. Lin, 2019 Hybrid regularized echo state network for multivariate chaotic time series prediction. IEEE Transactions on Cybernetics **49**: 2305–2315.

Yang, H., H. Ye, G. Wang, and T. Hu, 2005 Fuzzy neural very-short-term load forecasting based on chaotic dynamics reconstruction. In *Advances in Neural Networks – ISNN 2005*, edited by J. Wang, X.-F. Liao, and Z. Yi, pp. 622–627, Berlin, Heidelberg, Springer Berlin Heidelberg.

Yang, L., J. Zhang, X. Wu, Y. Zhang, and J. Li, 2016 A chaotic time series prediction model for speech signal encoding based on genetic programming. Applied Soft Computing **38**: 754–761.

Yang, X.-S. and S. Deb, 2010 Engineering optimisation by cuckoo search. International Journal of Mathematical Modelling and Numerical Optimisation **1**: 330–343.

Yong, Z., 2013 New prediction of chaotic time series based on local lyapunov exponent. Chinese Physics B **22**.

Yumei, Z., B. Shulin, L. Gang, and W. Xiaojun, 2019 Kernel estimation of truncated volterra filter model based on dfp technique and its application to chaotic time series prediction. Chinese Journal of Electronics **28**: 127–135.

Zhang, D., 2019a Wavelet transform. In *Fundamentals of Image Data Mining*, pp. 35–44, Springer.

Zhang, D. and M. Jiang, 2020 Hetero-dimensional multitask neuroevolution for chaotic time series prediction. IEEE Access **8**: 123135–123150.

Zhang, L., 2019b Evaluating the effects of size and precision of training data on ann training performance for the prediction of chaotic time series patterns. International Journal Of Software Science And Computational Intelligence-IJSSCI **11**: 16–30.

Zhang, L., F. Tian, S. Liu, L. Dang, X. Peng, *et al.*, 2013 Chaotic time series prediction of e-nose sensor drift in embedded phase space. Sensors And Actuators B-Chemical **182**: 71–79.

Zhang, M., B. Wang, Y. Zhou, and H. Sun, 2020 Woa-based echo state network for chaotic time series prediction. Journal of the Korean Physical Society **76**: 384–391.

Zhao, C., W. Ren, and M. Han, 2021 Adaptive sparse quantization kernel least mean square algorithm for online prediction of chaotic time series. Circuits Systems And Signal Processing **40**: 4346–4369.

Zhao, J., Y. Li, X. Yu, and X. Zhang, 2014 Levenberg-marquardt algorithm for mackey-glass chaotic time series prediction. Discrete Dynamics In Nature and Society **2014**.

Zheng, Y., S. Wang, J. Feng, and C. K. Tse, 2016 A modified quantized kernel least mean square algorithm for prediction of chaotic time series. Digital Signal Processing **48**: 130–136.

Zhongda, T., L. Shujiang, W. Yanhong, and S. Yi, 2017 A prediction method based on wavelet transform and multiple models fusion for chaotic time series. Chaos Solitons & Fractals **98**: 158–172.

Zhou, Y.-T., Y. Fan, Z.-Y. Chen, and J.-C. Sun, 2017 Multimodality prediction of chaotic time series with sparse hard-cut em learning of the gaussian process mixture model. Chinese Physics Letters **34**.

How to cite this article: Martínez-García, J. A., González-Zapata, A. M., Rechy-Rechy-Ramírez, E. J., and Tlelo-Cuautle, E. On the Prediction of Chaotic Time Series using Neural Networks. Chaos Theory and Applications, 4(2), 94-103, 2022.

CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.4 / **No.**2 / 2022 / **pp.**104-110
https://doi.org/10.51537/chaos.1112188

Stability Analysis of Bitcoin using Recurrence Quantification Analysis

Baki Unal*,[1]
*Iskenderun Technical University, 31200, Iskenderun, Hatay, Turkey.

ABSTRACT Cryptocurrencies are new kinds of electronic currencies based on communication technologies. These currencies have attracted the attention of investors. However, cryptocurrencies are very volatile and unpredictable. For investors, it is very difficult to make investment decisions in cryptocurrency market. Therefore, revealing changes in the dynamics of cryptocurrencies are valuable for investors. Bitcoin is the most popular and representative cryptocurrency in cryptocurrency market. In this study how dynamical properties of Bitcoin changed through time is analyzed with recurrence quantification analysis (RQA). RQA is a pattern recognition-based time series analysis method that reveals dynamics of the time series by calculating some metrics called RQA measures. This method has been successfully applied to nonlinear, nonstationary, short and chaotic time series and does not assume a statistical model. RQA can reveal important properties of time series data such as determinism, laminarity, stability, randomness, regularity and complexity. By using sliding window RQA we show that in 2021 RQA measures for Bitcoin prices collapse and Bitcoin becomes more unpredictable, more random, more unstable, more irregular and less complex. Therefore, dynamics and stability of the Bitcoin prices significantly changed in 2021.

KEYWORDS
Recurrence quantification analysis
RQA
Recurrence plot
Cryptocurrency market
Bitcoin

INTRODUCTION

In the age of information and communication, new digital currencies called cryptocurrencies have emerged (Härdle *et al.* 2020). These cryptocurrencies are operating without a central bank. The decentralized nature of these cryptocurrencies is the result of a technology called blockchain (Yuan and Wang 2018; Tredinnick 2019). These cryptocurrencies have received a lot of attention from investors. Therefore, it is important for investors to reveal the critical changes in the cryptocurrency market.

Cryptocurrency market is a self-organized complex system formed from complex network of traders (Aste 2019). Cryptocurrency prices are output of this complex system. Cryptocurrency prices exhibit high level of nonlinearity, uncertainty and volatility (Chaim and Laurini 2019; Alqaralleh *et al.* 2020). Therefore, prediction of cryptocurrency prices is very difficult (Mezquita *et al.* 2022). However, critical changes in cryptocurrency market can be diagnosed by using recurrence quantification analysis (RQA).

Manuscript received: 2 May 2022,
Revised: 15 June 2022,
Accepted: 15 July 2022.

[1] bakiunal@gmail.com, baki.unal@iste.edu.tr (**Corresponding Author**)

RQA is a pattern recognition-based time series analysis method which is applied to recurrence plots (RP). Theoretical background of RQA and RP methods is based on how the states of the system is repeated (recurred) during its time evolution. Both RP and RQA reveal the recurrence structure of states of the system in phase space. By analyzing these recurrence structures several judgements can be made on the dynamical properties of the system. A RP can be analyzed visually. In a RP vertical or diagonal lines or isolated points indicate different dynamical properties for the system. In RQA several metrics (measures) are calculated from a RP. These metrics are calculated from the lengths of the vertical and diagonal lines on a RP and reflects dynamical properties of the system. While the RP analysis is dependent on the subjective judgments of the observer, RQA presents a more objective analysis.

Bitcoin is the main cryptocurrency in the cryptocurrency market thus a representative cryptocurrency. In this study our main research question is how stability and dynamic properties of Bitcoin prices have changed during the period 17-08-2017 and 05-10-2021. To carry out this task we utilized sliding window RQA to demonstrate how RQA measures changes through time for Bitcoin. Since RQA measures reflects important characteristics of a time series such as determinism, predictability, randomness, laminarity, stability, regularity and complexity, changes in the RQA measures of Bitcoin reveal the changes of the such characteristics of the Bitcoin prices through time. Our findings reveal that dynamics and stability characteristics of Bitcoin prices significantly changed in 2021. Before 2021 Bitcoin prices quite stable. However, after 2021 Bitcoin prices becomes very unstable and unpredictable. Therefore, we can distinguish two periods in terms of stability for Bitcoin prices. Our main contribution to the literature is that by using RQA we diagnosed dynamical changes in the cryptocurrency market by using the representative cryptocurrency Bitcoin. As far as we know our study is the first study analyzing stability and dynamics of Bitcoin prices using RQA covering the period between 17-08-2017 and 05-10-2021.

Organization of our study is as follows. In the second part we review literature on sliding window RQA. In the third part the methodology is demonstrated. In the fourth part application and results are presented. In the last part study is concluded.

LITERATURE REVIEW

There is a comprehensive bibliography on recurrence plots (RPs), RQA and their applications at the Marwan *et al.* (2013) web site. In this work we utilized sliding window RQA methodology to reveal changes in dynamical properties of Bitcoin through time. In the literature there are few studies applied sliding window RQA to financial time series. Bastos and Caiado (2011) applied RQA to daily data of 23 developed and 23 emerging stock markets between the dates January 1995 and December 2009. By using sliding window RQA, authors demonstrated that during critical economic events such as dot-com bubble, Asian financial crisis and 2008 subprime mortgage crisis, RQA measures laminarity (*LAM*) and determinism (*DET*) decline.

Piskun and Piskun (2011) investigated several stock market crashes by using sliding window RQA and show that RQA measure *LAM* can be used to identify market bubbles. Authors demonstrated that *LAM* measure can be used to distinguish different market periods such as normal functioning, instability, critical period and relaxation.

Sasikumar and Kamaiah (2014) analyzed two Indian stock market indices between 2 January 2002 and 10 October 2013 with sliding window RQA. Authors concluded that Indian equity market has chaotic nature. Also, they demonstrated that RQA measure determinism collapse during the 2008 subprime mortgage crisis and 2010 Euro zone debt crisis. The authors concluded that after 2008 subprime mortgage crisis the market was in turbulent state. Additionally, authors investigated the change in RQA measure *LAM* through time. They showed that laminarity collapsed during the 2008 subprime mortgage crisis. Authors' results for RQA measure trapping time (*Vmean*) confirm that after 2008 subprime mortgage crisis market becomes turbulent.

Moloney and Raghavendra (2012) utilized sliding window RQA to analyze the transition of Dow Jones Industrial Index from bull market to bear market. Authors have particularly interested with events of peaks and subsequent crashes in the dates 1929, 1973, 2000, 2007. Authors discovered that the RQA measures fall soon before or around market peaks. This means that around market peaks, dynamics of the market lose its deterministic structure. Authors detected phase transitions when market transforms from bull state to bear state.

Soloviev *et al.* (2020) analyzed 9 critical periods in the Dow Jones Industrial Average (DJIA) index for the period between 1 January 1990 and 1 June 2019 by using sliding window RQA. Authors demonstrated that during all critical periods RQA measure *DET* shows a downward trend and can detect critical phenomenon. They indicated that *DET*, *LAM*, longest diagonal line (*Lmax*) and trapping time (*Vmean*) are the RQA measures most sensitive to critical events.

Soloviev and Belinskiy (2018) demonstrated possibility of constructing indicators of critical and crisis events in Bitcoin prices using RQA. Authors used daily Bitcoin prices covering the period between 16 July 2010 and 10 February 2018. Authors concluded that RQA measures such as recurrence rate (*REC*), determinism (*DET*) and entropy (*ENTR*) are excellent candidate for a fast, robust, and useful screener and detector of unusual patterns in complex time series.

Soloviev and Belinskiy (2019) used complexity measures to investigate crashes and critical phenomena in the cryptocurrency market. Authors showed that before the crashes and the actual periods of crashes complexity of the market system changes.

In the literature there are few studies applying RQA methodology to Bitcoin. These are Soloviev and Belinskiy (2018, 2019), Kucherova *et al.* (2021) and Bielinskyi and Serdyuk (2021). However, focus of these studies is not to evaluate the dynamical stability of the Bitcoin and these studies do not evaluate the full spectrum of RQA measures but only consider a small subset of RQA measures. Also, data utilized in these studies do not cover recent 2021 data. Focus of Soloviev and Belinskiy (2018, 2019) and Bielinskyi and Serdyuk (2021) is to evaluate the suitability of RQA measures as precursors of crisis and crashes in cryptocurrency market. Focus of Kucherova *et al.* (2021) is to reveal the relationship between the time series of the price of Bitcoin and the frequency of online requests for Bitcoin. The authors used this relationship to illuminate the behavior of agents in the digital economy. After all, as far as we know our study is the first study investigating stability and dynamic properties of Bitcoin prices using broad spectrum of RQA measures and up to date 2021 data.

METHODOLOGY

Recurrence plots (RPs) (Packard *et al.* 1980; Takens 1981; Eckmann *et al.* 1987) are visual analysis tools which portray repetitions of the states of the time series. By visual inspection of RPs dynamics of the underlying time series can be identified. However visual inspection of the RPs has some limitations such as subjective judgement of the observer. To overcome these limitations recurrence quantification analysis (RQA) is developed (Zbilut and Webber 1992; Webber Jr and Zbilut 1994; Marwan *et al.* 2002). In RQA simple pattern recognition algorithms are applied to a RP and measures that describe various properties of the time series are obtained. These analysis tools are successfully applied to nonlinear, nonstationary and chaotic time series in the literature. Main advantages of these tools are that they do not require assumptions such as stationarity, statistical distributions or necessary number of observations. A RP can be expressed by following formula:

$$RP_{ij} = \Theta(T - \|V_i(x) - V_j(x)\|) \tag{1}$$

In the expression above Θ denotes Heaviside step function and T denotes threshold value. If the distance between two state vectors is lower than a threshold value corresponding elements of the recurrence matrix takes the value of one.

In RPs adjacent points have a special meaning. When adjacent points form diagonal lines, this means that states visit same region at different times. Length of these diagonal lines reflects duration of these visits. When adjacent points form vertical or horizontal lines, this means states stay in same region for a duration. RPs belong to deterministic systems display long diagonal lines and few isolated points and RPs belong to stochastic systems display isolated points or very short diagonal lines. In RQA following measures can be calculated:

REC (recurrence rate) quantifies fraction of points in the RP. This metric reflects the likelihood of recurrence of a state. *REC* can be calculated by the following formula:

$$REC = \frac{1}{N^2} \sum_{i,j=1}^{N} RP(i,j) \tag{2}$$

In the formula above, N denotes the number of points on the constructed phase space.

DET (determinism) quantifies fraction of points in the RP which forms diagonal lines. This metric reflects determinism and randomness in the system. *DET* can be calculated by the following formula:

$$DET = \frac{\sum_{l=l_{min}}^{N} lP(l)}{\sum_{l=1}^{N} lP(l)} \tag{3}$$

In the formula above $P(l)$ represents the frequency distribution of the diagonal lines with length l.

Lmax is the longest diagonal line's length. This metric reflects the stability of the system. High *Lmax* value means high stability and low *Lmax* value means low stability. This metric is also inversely related with largest positive Lyapunov exponent. Lmax can be calculated by the following formula:

$$Lmax = \max(\{l_i; i = 1, \ldots, N_l\}) \tag{4}$$

In the formula above N_l represents the number of diagonal lines in the RP.

ENTR is the Shannon entropy of the diagonal line length distribution. This metric reflects the diversity and the complexity of diagonal lines. A high *ENTR* value means complexity is high and a low *ENTR* value means complexity is low. *ENTR* value can be obtained from following formula:

$$ENTR = -\sum_{l=l_{min}}^{N} p(l) \ln p(l) \tag{5}$$

In the formula above $p(l)$ denotes probability of a diagonal line has length l.

LAM (laminarity) quantifies fraction of points in the RP which forms vertical lines. This metric reflects laminar states in the system. A higher *LAM* values mean higher regularity in the system. This measure can detect chaos-chaos transitions. *LAM* value can be calculated by the following formula:

$$LAM = \frac{\sum_{v=v_{min}}^{N} vP(v)}{\sum_{v=1}^{N} vP(v)} \tag{6}$$

In the formula above $P(v)$ denotes frequency distribution of vertical lines with length v.

Vmean is the average length of vertical lines. This metric reflects the average trapping time of the system in particular states. *Vmean* can be calculated by the following formula:

$$Vmean = \frac{\sum_{v=v_{min}}^{N} vP(v)}{\sum_{v=v_{min}}^{N} P(v)} \tag{7}$$

Lmean is average length of the diagonal lines. It is the average amount of time that the two segments of the trajectory are in close proximity to one another. It can be considered as average time for forecast. *Lmean* can be calculated by the following formula:

$$Lmean = \frac{\sum_{l=l_{min}}^{N} lP(l)}{\sum_{l=l_{min}}^{N} P(l)} \tag{8}$$

APPLICATION AND RESULTS

In this study hourly prices of Bitcoin between dates 17 August 2017 and 5 October 2021 are used. Data is obtained from the cryptocurrency market Binance. In this study also sliding window methodology is adopted. Window size is selected as 1000 and window step size for sliding is selected as 200. The calculations were performed using the *nonlinearTseries* package of the R software.

First step in the analysis of a time series with a RP and RQA is embedding the original univariate time series to obtain multidimensional state vectors. In this procedure a univariate time series such as **x** (9) is converted to multivariate time series such as **V** (10). This procedure is called phase space reconstruction. In this procedure two parameters must be defined. These are embedding dimension (D) and time delay (τ). To determine these parameters, methods such as false nearest neighbors and mutual information are suggested (Huffaker *et al.* 2017).

However, Zbilut (2005) asserted that for economic time series embedding dimension can be selected as 10 and time delay can be selected as 1. In this study we followed Zbilut (2005) suggestion and selected embedding parameters likewise. Percentage of false nearest neighbors graph is presented in Figure 1. From this figure it is seen that after the embedding dimension of 10, percent of false nearest neighbors does not decline much. Therefore, selecting embedding dimension as 10 is an appropriate choice. Average mutual information graph is presented in Figure 2. In the literature it is recommended to choose time delay as the first local minimum of the average mutual information graph. As seen from Figure 2 there is no local minimum. Therefore, by following suggestion of Zbilut (2005) we set time delay as 1 and do not skip any observation. This parameter setting is coherent with the works of Strozzi *et al.* (2007), Strozzi *et al.* (2008), Bastos and Caiado (2011) and Xing and Wang (2020).

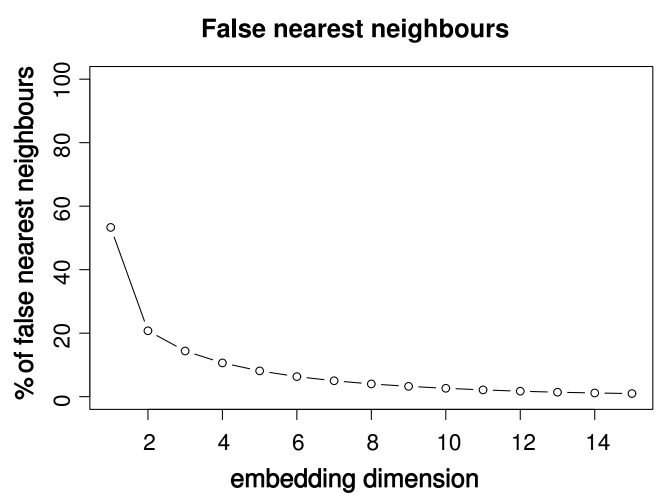


Figure 1 Percentage of false nearest neighbors

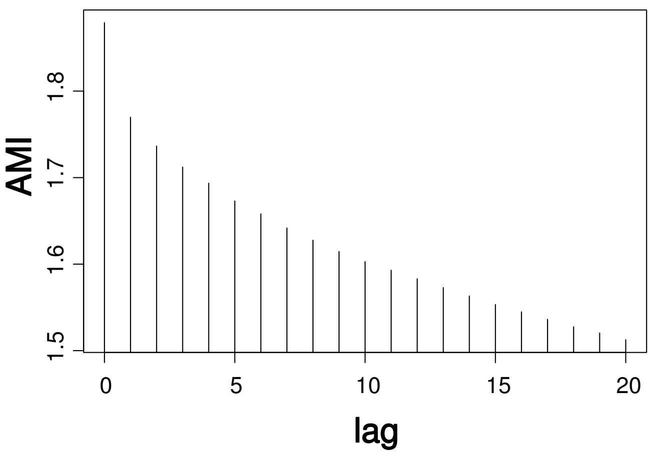


Figure 2 Average mutual information

$$\mathbf{x} = (x_1, x_2, x_3, \dots, x_n) \tag{9}$$

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \vdots \\ \mathbf{V}_{n-(D-1)\tau} \end{pmatrix} \tag{10}$$

$$\mathbf{V} = \begin{pmatrix} x_1 & x_{1+\tau} & \dots & x_{1+(D-1)\tau} \\ x_2 & x_{2+\tau} & \dots & x_{2+(D-1)\tau} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-(D-1)\tau} & x_{n-(D-2)\tau} & \dots & x_n \end{pmatrix} \tag{11}$$

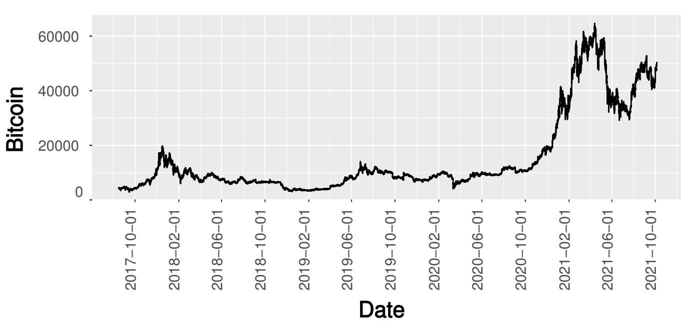


Figure 3 Hourly Bitcoin prices

Graph of hourly Bitcoin prices are presented in Figure 3. As seen from this there is a sharp increase of Bitcoin prices in 2021. Also, Bitcoin prices become more volatile in that time. Changes in the RQA measure recurrence rate (*REC*) through time is presented in Figure 4. As seen from this figure recurrence rate exhibit a fluctuating pattern until 2021. However, in 2021 recurrence rate

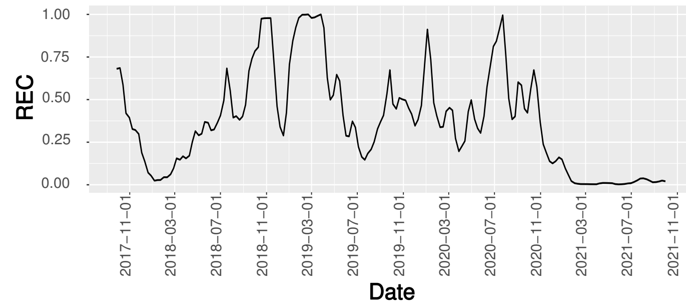


Figure 4 Changes in recurrence rate (REC)

collapse. This means repetitions of the states are significantly reduced in 2021.

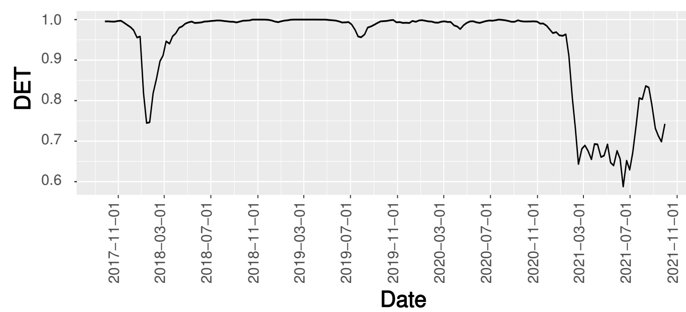


Figure 5 Changes in determinism (DET)

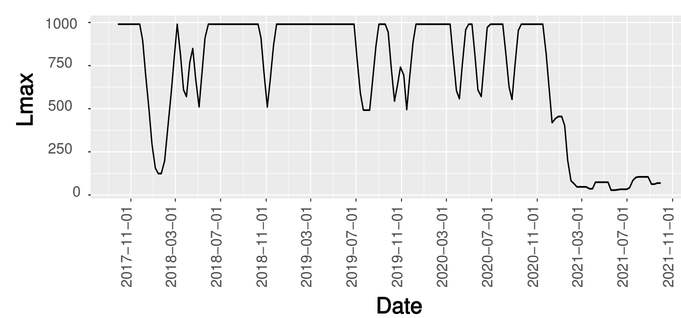


Figure 6 Changes in the longest diagonal line (Lmax)

Changes in the RQA measure determinism (*DET*) through time is depicted in Figure 5. In this figure there is a local dip in determinism in the beginning of the 2018. However more noticeably there is a collapse in determinism in 2021. Since RQA measure determinism reflects the predictability and the randomness of the time series this collapse means Bitcoin becomes more unpredictable and more random in 2021. In the Figure 6 how RQA measure the longest diagonal line (*Lmax*) changes through time is demonstrated. In this Figure until 2021 several local dips are observed. But in 2021 there is a total collapse in Lmax values. Since Lmax reflects the stability of the dynamics and related inversely with largest positive Lyapunov exponent, collapse in 2021 reflects that stability of Bitcoin is significantly reduced in 2021. Changes in the RQA measure laminarity (*LAM*) through time is shown in Figure 7. This figure is similar to the Figure 5 for determinism. In the Figure 7 there is also a local dip in the beginning of the 2018. However, laminarity collapse in 2021 similar to determinism.

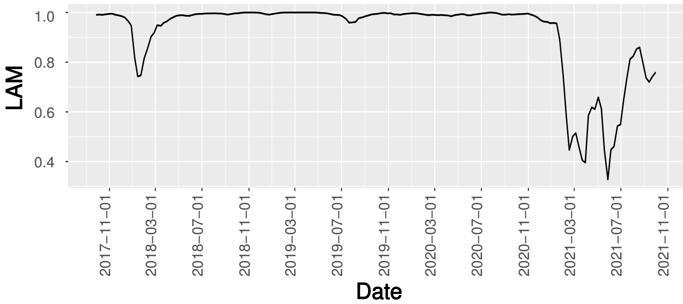


Figure 7 Changes in the laminarity (LAM)

Since RQA measure *LAM* is sensitive to critical changes in the dynamics, collapse in 2021 indicates that dynamics of the Bitcoin is substantially changed and Bitcoin entered a critical state in 2021. Changes in the RQA measure mean length of the diagonal lines (*Lmean*) is presented in Figure 8. This RQA measure gauge the average time for forecast. From this figure it is seen that average time for forecast is greatly reduced in 2021. This again confirms that predictability of the Bitcoin is reduced in 2021.

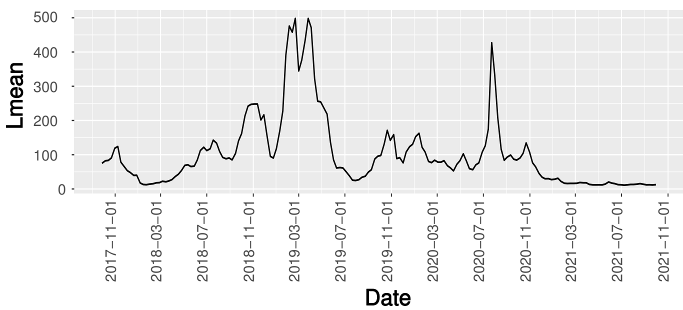


Figure 8 Changes in mean length of the diagonal lines (Lmean)

In Figure 9 changes in the RQA measure average length of vertical lines (*Vmean*) are shown. This RQA measure gauge average trapping time of the system in particular states. From this figure it can be seen that trapping time in particular states are significantly reduced and transitions between states are accelerated in 2021. Changes in the RQA measure Shannon entropy (*ENTR*) is depicted in Figure 10. In this figure it is seen that Shannon entropy is collapsed in 2021. Since Shannon entropy reflects the complexity of the system this collapse reflects that complexity of the Bitcoin is reduced in 2021.

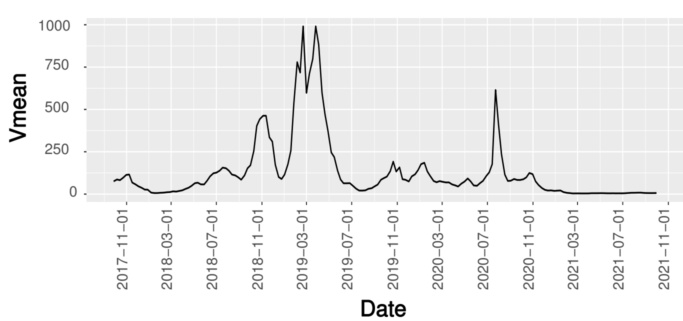


Figure 9 Changes in average length of vertical lines (Vmean)

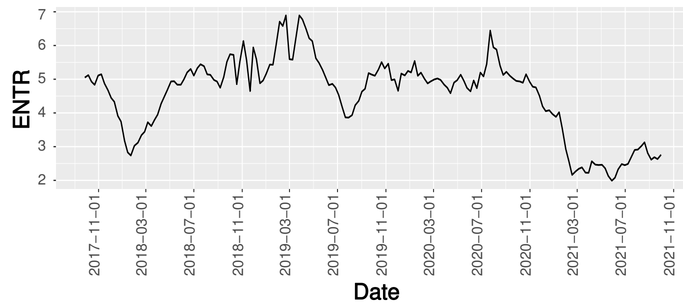


Figure 10 Changes in Shannon entropy (ENTR)

To facilitate ease of comparison, in Figures 11-14 Bitcoin prices and RQA measures are shown on the same graphs. In these figures red curves represents Bitcoin prices and blue curves represents RQA measures. To make a meaningful comparison, Bitcoin prices, *Lmax* and *ENTR* are normalized to the range between 0 and 1. Since *DET* and *LAM* measures take values between 0 and 1, normalization is not required for these variables. As seen from Figures 11-14 the increase in Bitcoin prices is accompanied by a decrease in RQA measures. These graphs reveal that Bitcoin price dynamics are significantly changed in 2021.

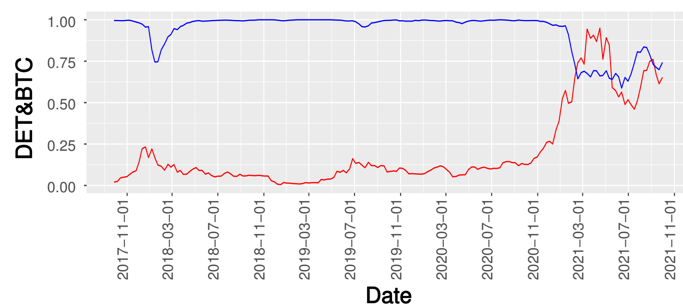


Figure 11 Determinism vs. Bitcoin prices. Red curve denotes Bitcoin prices and blue curve denotes determinism

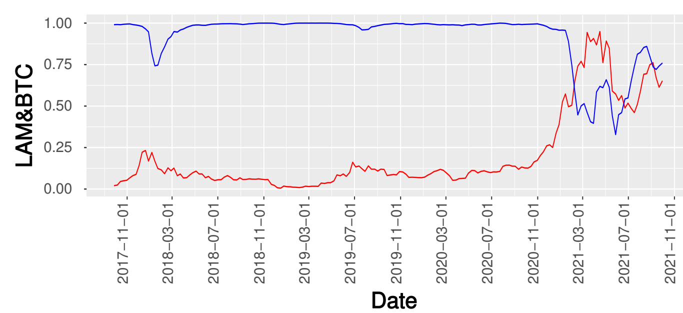


Figure 12 Laminarity vs. Bitcoin prices. Red curve denotes Bitcoin prices and blue curve denotes laminarity

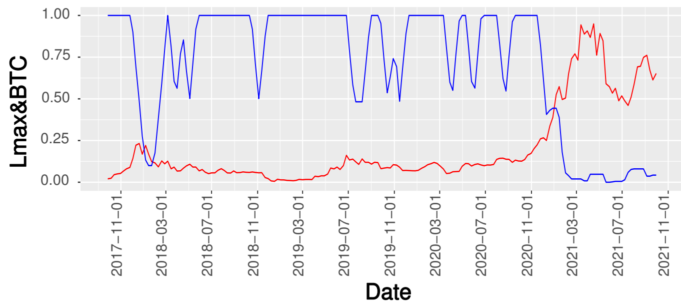


Figure 13 Longest diagonal line's length (Lmax) vs. Bitcoin prices. Red curve denotes Bitcoin prices and blue curve denotes Lmax

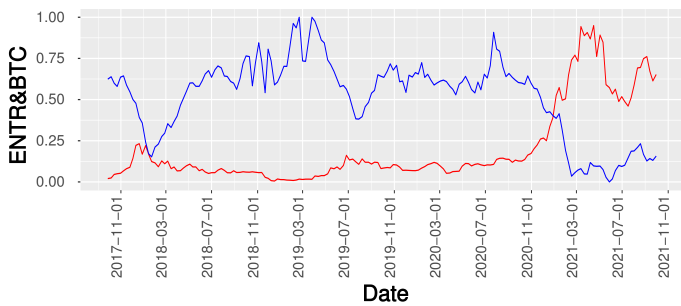


Figure 14 Shannon entropy vs. Bitcoin prices. Red curve denotes Bitcoin prices and blue curve denotes Shannon entropy

CONCLUSION

In this study recurrence quantification analysis is applied to Bitcoin prices to reveal how dynamic properties and stability of Bitcoin prices changed through time. In this analysis change in RQA measures are demonstrated by using sliding window methodology. In the literature it is shown that during or at the beginning of the critical periods such as crisis RQA measures collapse. In this study we demonstrated that RQA measures for Bitcoin prices collapsed in 2021. This means Bitcoin prices become more unpredictable, more random, more unstable, more irregular and less complex in 2021.

Therefore, stability and dynamic characteristics of Bitcoin have been significantly changed in 2021. From this analysis we also can distinguish two different periods for Bitcoin, namely stable and unstable periods. Period before 2021 can be labelled as stable period and period after 2021 can be labelled as unstable period. Therefore, Bitcoin enters a state of turbulence in 2021. From the investors' point of view this means that making investment decisions for Bitcoin becomes much more difficult in 2021. So, what is the reason for this change? Further studies are required to answer this question. Possible explanations can be increase in transaction volumes in the cryptocurrency markets, changes in traders' behaviors, changes in the market conditions and the COVID-19 pandemic.

For traditional currencies when a currency become unstable the corresponding central bank intervenes in the market to stabilize the currency. However, for Bitcoin there is no central bank which decides the amount of emission of the currency. Therefore, since there is no policy maker for Bitcoin, we have no policy implications for Bitcoin. However, we have some implications for investors. Since Bitcoin lost its deterministic structures in 2021 mathematical and statistical models which explain future Bitcoin prices with past realizations become infeasible. Therefore, mathematical models such as difference and differential equations or statistical models such as ARIMA become unsuitable for forecasting future bitcoin prices using past bitcoin prices in 2021. Therefore, investors should consider this situation when creating their investment strategies.

Conflicts of interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Availability of data and material

Not applicable.

LITERATURE CITED

Alqaralleh, H., A. A. Abuhommous, A. Alsaraireh, *et al.*, 2020 Modelling and forecasting the volatility of cryptocurrencies: A comparison of nonlinear garch-type models. International Journal of Financial Research **11**: 346–356.

Aste, T., 2019 Cryptocurrency market structure: connecting emotions and economics. Digital Finance **1**: 5–21.

Bastos, J. A. and J. Caiado, 2011 Recurrence quantification analysis of global stock markets. Physica A: Statistical Mechanics and its Applications **390**: 1315–1325.

Bielinskyi, A. and O. Serdyuk, 2021 Econophysics of cryptocurrency crashes: an overview .

Chaim, P. and M. P. Laurini, 2019 Nonlinear dependence in cryptocurrency markets. The North American Journal of Economics and Finance **48**: 32–47.

Eckmann, J.-P., S. O. Kamphorst, and D. Ruelle, 1987 Recurrence plots of dynamical systems. Europhysics Letters (EPL) **4**: 973–977.

Härdle, W. K., C. R. Harvey, and R. C. G. Reule, 2020 Understanding cryptocurrencies. Journal of Financial Econometrics **18**: 181–208.

Huffaker, R., R. G. Huffaker, M. Bittelli, and R. Rosa, 2017 *Nonlinear time series analysis with R*. Oxford University Press.

Kucherova, H. Y., V. O. Los, D. V. Ocheretin, O. V. Bilska, and E. V. Makazan, 2021 Innovative behavior of bitcoin market agents during covid-19: recurrence analysis. In *M3E2-MLPEED*, pp. 1–15.

Marwan, N., N. Wessel, U. Meyerfeldt, A. Schirdewan, and J. Kurths, 2002 Recurrence-plot-based measures of complexity and their application to heart-rate-variability data. Phys. Rev. E **66**: 026702.

Marwan, N., N. Wessel, U. Meyerfeldt, A. Schirdewan, and J. Kurths, 2013 A comprehensive bibliography about rps, rqa an their applications.

Mezquita, Y., A. B. Gil-González, J. Prieto, and J. M. Corchado, 2022 Cryptocurrencies and price prediction: A survey. In *Blockchain and Applications*, edited by J. Prieto, A. Partida, P. Leitão, and A. Pinto, pp. 339–346, Cham, Springer International Publishing.

Moloney, K. and S. Raghavendra, 2012 Examining the dynamical transition in the dow jones industrial. Physics Letters A **223**: 255–260.

Packard, N. H., J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, 1980 Geometry from a time series. Phys. Rev. Lett. **45**: 712–716.

Piskun, O. and S. Piskun, 2011 Recurrence quantification analysis of financial market crashes and crises .

Sasikumar, A. and B. Kamaiah, 2014 A complex dynamical analysis of the indian stock market. Economics Research International **2014**.

Soloviev, V. and A. Belinskiy, 2018 Methods of nonlinear dynamics and the construction of cryptocurrency crisis phenomena precursors .

Soloviev, V., O. Serdiuk, S. Semerikov, and A. Kiv, 2020 Recurrence plot-based analysis of financial-economic crashes. CEUR Workshop Proceedings.

Soloviev, V. N. and A. Belinskiy, 2019 Complex systems theory and crashes of cryptocurrency market. In *Information and Communication Technologies in Education, Research, and Industrial Applications*, edited by V. Ermolayev, M. C. Suárez-Figueroa, V. Yakovyna, H. C. Mayr, M. Nikitchenko, and A. Spivakovsky, pp. 276–297, Cham, Springer International Publishing.

Strozzi, F., E. Gutiérrez, C. Noè, T. Rossi, M. Serati, *et al.*, 2008 Measuring volatility in the nordic spot electricity market using recurrence quantification analysis. The European Physical Journal Special Topics **164**: 105–115.

Strozzi, F., J.-M. Zaldívar, and J. P. Zbilut, 2007 Recurrence quantification analysis and state space divergence reconstruction for financial time series analysis. Physica A: Statistical Mechanics and its Applications **376**: 487–499.

Takens, F., 1981 Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980*, edited by D. Rand and L.-S. Young, pp. 366–381, Berlin, Heidelberg, Springer Berlin Heidelberg.

Tredinnick, L., 2019 Cryptocurrencies and the blockchain. Business Information Review **36**: 39–44.

Webber Jr, C. L. and J. P. Zbilut, 1994 Dynamical assessment of physiological systems and states using recurrence plot strategies. Journal of Applied Physiology **76**: 965–973, PMID: 8175612.

Xing, Y. and J. Wang, 2020 Linkages between global crude oil market volatility and financial market by complexity synchronization. Empirical Economics **59**: 2405–2421.

Yuan, Y. and F.-Y. Wang, 2018 Blockchain and cryptocurrencies: Model, techniques, and applications. IEEE Transactions on Systems, Man, and Cybernetics: Systems **48**: 1421–1428.

Zbilut, J. P., 2005 Use of recurrence quantification analysis in economic time series. In *Economics: Complex Windows*, pp. 91–104, Springer.

Zbilut, J. P. and C. L. Webber, 1992 Embeddings and delays as derived from quantification of recurrence plots. Physics Letters A **171**: 199–203.

How to cite this article: Unal, B. Stability Analysis of Bitcoin using Recurrence Quantification Analysis. Chaos Theory and Applications, 4(2), 104-110, 2022.