Received: 30 October 2022 | Revised: 19 February 2023 | Accepted: 12 March 2023

DOI: 10.1002/mma.9228

RESEARCH ARTICLE

WILEY

Recovering discrete delayed fractional equations from trajectories

J. Alberto Conejero[1] | **Òscar Garibo-i-Orts**[1,2] | **Carlos Lizama**[3]

[1]Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València, 46022 Valencia, Spain

[2]GRID—Grupo de Investigación en Ciencia de Datos, Valencian International University—VIU, 46002 Valencia, Spain

[3]Departamento de Matemática y Ciencia de la Computación, Universidad de Santiago de Chile, Santiago, Chile

Correspondence
Òscar Garibo-i-Orts, GRID—Grupo de Investigación en Ciencia de Datos, Valencian International University—VIU, Carrer Pintor Sorolla 21, 46002 Valencia, Spain.
Email: oscar.garibo@campusviu.es

Communicated by: N. Hyvonen

Funding information
Ministerio de Ciencia e Innovación—Agencia Estatal de Investigación, Grant/Award Number: PID2021-124618NB-C21; European Union

We show how machine learning methods can unveil the fractional and delayed nature of discrete dynamical systems. In particular, we study the case of the fractional delayed logistic map. We show that given a trajectory, we can detect if it has some delay effect or not and also to characterize the fractional component of the underlying generation model.

KEYWORDS

chaotic systems, delayed discrete fractional systems, fractional dynamical systems, machine learning, recurrent neural networks

MSC CLASSIFICATION

26A33, 65P20, 68U20

1 | INTRODUCTION

Discrete fractional calculus (DFC) permits a description of dynamical systems of real-world processes with discrete nature where there exist long-term connections between different time steps. Such connections are introduced as discrete-memory effects in the model, and DFC has been shown a reliable approach in order to capturing them. Recent examples can be found in the study of chaos [1], variable-order discrete-time recurrent neural networks [2], and fuzzy systems on discrete-time domains [3]. As a matter of fact, there have been found interesting applications in different settings such as image encryption [4–7] or earthquakes intensity modeling [8–10].

The origins of DFC are dated to 1956, when Kutter [11] mentioned for the first time differences of fractional order. In 1974, Diaz and Osler [12] introduced a discrete fractional difference operator defined as an infinite series. In 1988, Gray and Zhang [13] developed a fractional calculus for the discrete ∇ (backward) difference operator. Miller and Ross [14] defined a fractional sum via the solution of a linear difference equation. Their definition is the discrete analog of the Riemann–Liouville fractional integral, which can be obtained via the solution of a linear differential equation. In 2007, Atici and Eloe [15] introduced the Riemann–Liouville-like fractional difference by using the definition of a fractional sum of Miller and Ross and developed some of its properties that allow one to obtain solutions of certain fractional difference equations. The presence of chaos in these models was first studied for the logistic map [1]; see also Wu et al. [16] for sine maps.

Later, Wu and Baleanu also studied the chaos in a delayed version of the logistic map. They started from an expression where the forward Euler operator Δ was equal to the nonlinear right term of the logistic, that is, $\Delta x(n) := x(n+1) - x(n)$,

and then, they replaced the Euler operator with the left Caputo discrete difference operator Δ^{α}, obtaining

$$x(n) = x(0) + \frac{\mu}{\Gamma(\nu)} \sum_{j=1}^{n} \frac{\Gamma(n-j+\nu)}{\Gamma(n-j+1)} x(j-1)(1-x(j-1)), \tag{1}$$

where μ is a parameter and ν is a scaling factor. Such a relationship can also be obtained by convolution, using the Cesàro numbers of order ν, $k^{\nu}(j) = \frac{\Gamma(\nu+j)}{\Gamma(\nu)\Gamma(j+1)}$ with $j \in \mathbb{N}_0$, as a memory kernel in terms of the scaling factor ν [17]. It is worth mentioning that new fractional difference equations have been introduced with a general logarithm function on time scales as a kernel function. Such equations are known as Caputo–Hadamard fractional difference equations [18, 19].

After their first work on discrete fractional chaos, Wu and Baleanu later introduced a two-dimensional dynamical system including a delay term [20] that reads as

$$\begin{aligned} x(n) &= x(0) + \frac{\mu}{\Gamma(\nu)} \sum_{j=1}^{n} \frac{\Gamma(n-j+\nu)}{\Gamma(n-j+1)} x(j-1)(1-y(j-1)) \\ y(n) &= x(n-1). \end{aligned} \tag{2}$$

This delayed fractional discrete dynamical system also presents chaos as can be noticed in the Feigenbaum diagrams computed in terms of the parameter μ. As a matter of fact, in Figures 1 and 2, we show the Feigenbaum diagrams computed for the initial conditions $x(0) = y(0) = 0.3$ and $x(0) = y(0) = 0.8$.

The emergence of machine learning as a research tool has been extended to almost all research fields. It has shown its potential in Physics being used for studying anomalous diffusion noisy trajectories, which are characterized as the ones whose variance of the mean square displacement grows with respect to the time in terms of t^{α}, with α being the fractional diffusion exponent [21]. Examples of models generating those types of trajectories are annealed transient time motion (ATTM) [22], continuous time random walk (CTRW) [23], fractional Brownian motion (FBM) [24, 25], Lévy walks (LW) [26, 27], and scaled Brownian motion (SBM) [28].

Several models have been developed in the frame of the Andi Challenge for inferring the fractional exponent α and for determining the generating model of 1D, 2D, and 3D noisy trajectories; see, for instance, previous studies [29–31]. We refer the reader for a full comparison between these models to Muñoz-Gil et al. [27].

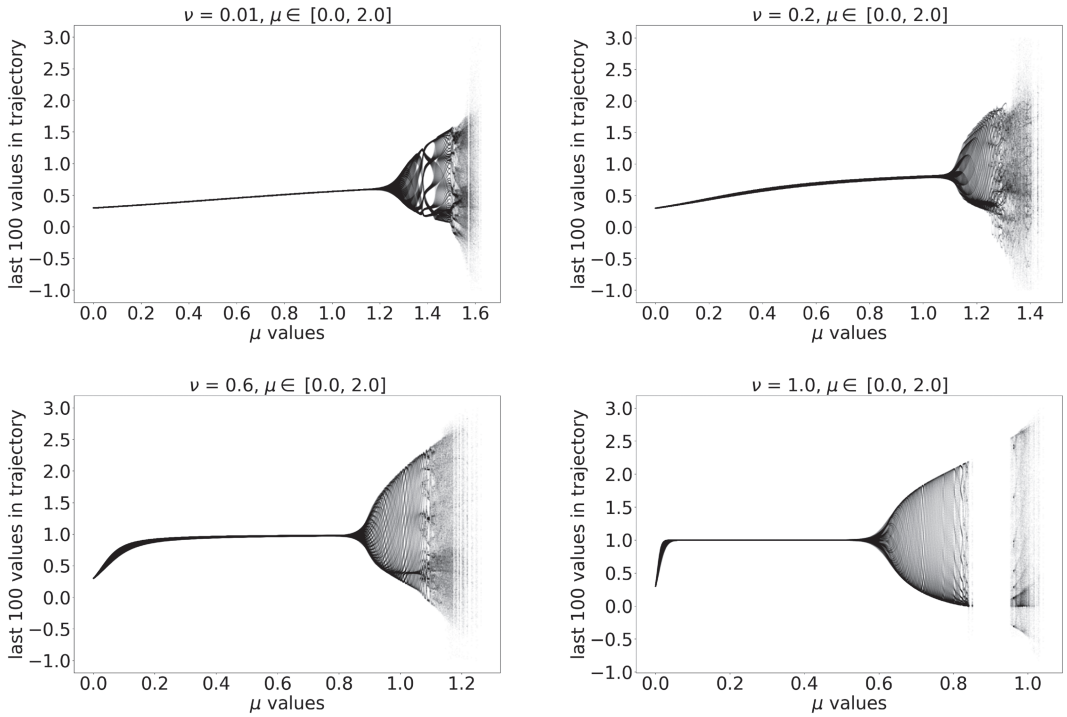

FIGURE 1 Feigenbaum plots for the dynamical system given by (1) for $x(0) = y(0) = 0.3$ and $\nu = 0.01$ (top left), $\nu = 0.2$ (top right), $\nu = 0.6$ (bottom left), and $\nu = 1$ (bottom right). For each value μ, we compute 200 terms of the sequence, and we plot the last 100 values.

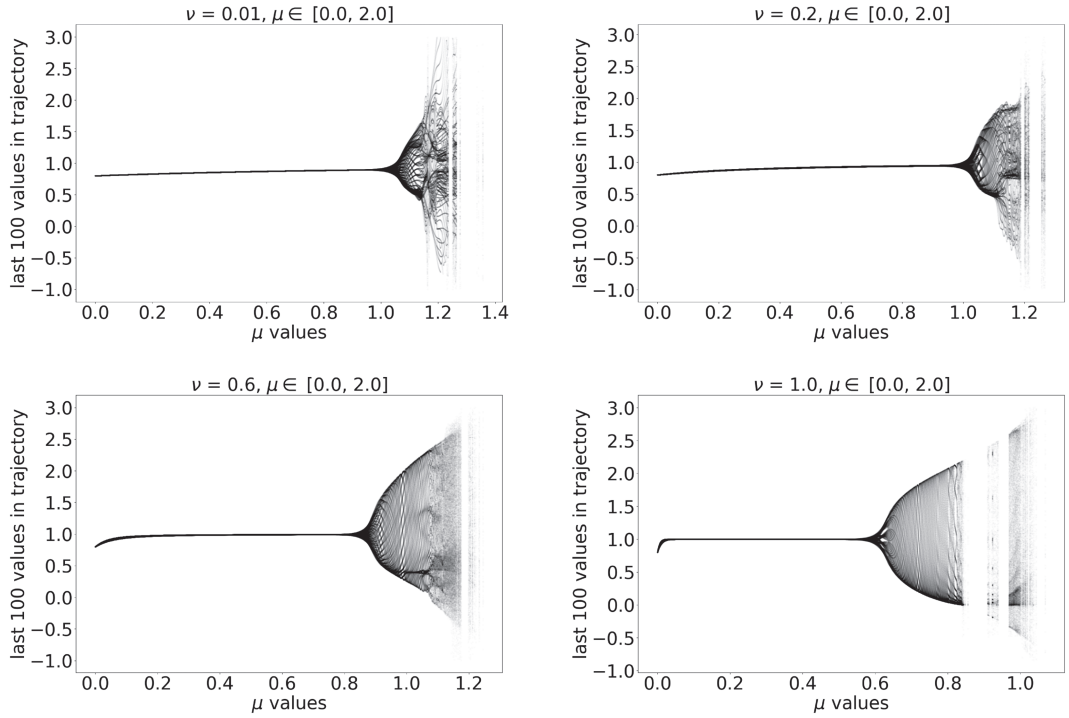

FIGURE 2 Feigenbaum plots for the dynamical system given by (1) for $x(0) = y(0) = 0.8$ and $\nu = 0.01$ (top left), $\nu = 0.2$ (top right), $\nu = 0.6$ (bottom left), and $\nu = 1$ (bottom right). For each value μ, we compute 200 terms of the sequence, and we plot the last 100 values.

Machine learning and artificial intelligence methods have been also successfully incorporated in the study of formal mathematical problems, as is the case of finding the solution of nonlinear models [32, 33] or the recent success of discovering new multiplication algorithms [34].

Fractional models are suitable for representing dynamical systems with a memory effect, either in space or time, due to the non-local character of fractional operators, as it can be shown in the following examples [35–38]. There is not a universal definition of fractional derivative [39], and in some cases, it is not straightforward to know which definition would be better to choose [40].

Recently, we have studied how machine learning models can be used in connection with fractional models in order to infer the μ parameter and the scaling factor ν of the fractional version of the logistic equation [41]. In this work, we study up to which point we can predict the μ and ν parameters from short trajectories generated by the discrete fractional delayed dynamical system given by (2). We also analyze if we can discriminate if trajectories contained a delayed component by comparing trajectories generated by the aforementioned systems described in (2) and (1).

For this purpose, we have used a model that combines convolutional and recurrent neural networks, that presented successful results for inferring the exponent α of anomalous diffusion trajectories [30] and for predicting the number of new COVID19 cases [42]. We will show in which cases we can clearly determine from a given trajectory (1) which were the parameters used for generating them and (2) if we can affirm that it contains a delay component or not.

In Section 2, we briefly outline the architecture of the machine learning model used. We also show how we construct the training, validation, and test data sets. We introduce the results in Section 3, and we draw some conclusions in Section 4.

2 | METHODOLOGY

We start by describing our network architecture we depict in Figure 3 and that consists of three parts:

1. First, a trajectory is processed by two convolutional layers. They are intended for feature extraction since they retain spatial structure inherent to trajectories by setting a patch, which behaves as a sliding window, that goes along the complete trajectory, connecting each patch of size kernel size, to a single neuron. By doing so, different neurons specialize in different regions in the trajectory [43, 44].

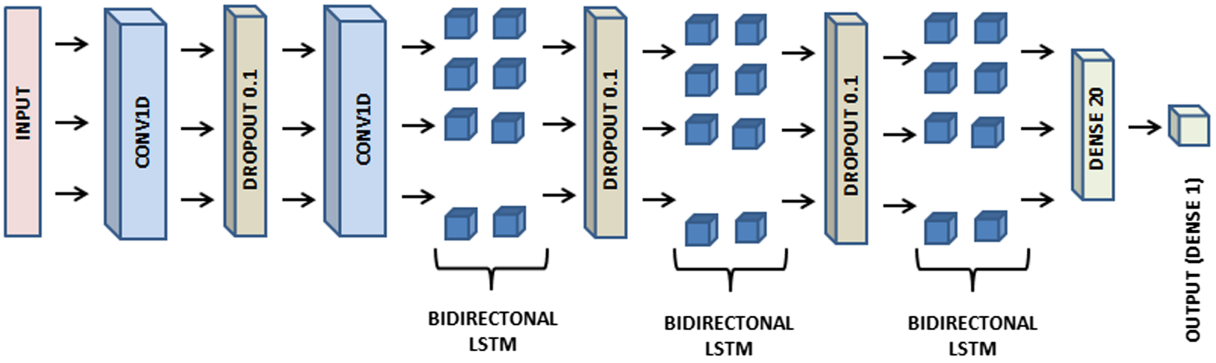

FIGURE 3 Machine learning architecture used for inferring the generating μ and ν parameters of the delayed logistic model trajectories. [Colour figure can be viewed at wileyonlinelibrary.com]

The first convolutional layer consists of 32 filters, and the second one of 64, with a sliding window (kernel) of size 5, following the doubling rule [43, 44]. The different number of filters returns features at different scales while retaining the spatial structure.

2. Second, the features feed 3 stacked bidirectional long short-term memory (LSTM) of 64 inputs and 32 units each one [45, 46] using Adam optimizer [47]. This type of recurrent neural network is able to learn sequential dependencies from the extracted features. For this purpose, LSTM layers include gated cells to avoid short-term memory or gradient vanishing, allowing the layer to learn from the complete trajectory, even from positions at the beginning of it. A cell state channel is selectively updated to allow getting rid of irrelevant information while retaining the important pieces.

   In other words, they are able to retain memory about early positions in the trajectory, thus considering the complete trajectory while adjusting to the target (μ and ν). We also include a dropout layer after each of these LSTM of the 10% neurons to avoid over-fitting.
3. Lastly, the output of the LSTM blocks is passed to two dense layers of 20 and 1 unit. This last choice depends if we want to predict a single parameter or both of them at the same time. The activation function of this last dense layer depends on the problem to be addressed: It is a linear function for regression of the μ and ν parameters and a sigmoid function for determining whether a trajectory comes from a delayed model or not, that is, for the classification problem.

In order to train this model, we built two separate data sets for training and validating while training, the previous model. Once the model was trained, we also constructed an additional data set for testing. These data sets are generated according to the following indications:

- $\mu \in [0.0, 2.0]$ with increments of 0.001.
- $\nu \in [0.01, 1]$ with increments of 0.01.
- trajectory length N, with $N \in [10,50]$ randomly selected.
- $u_0 \in [0, 1]$ randomly chosen with a resolution of 10^{-2}.

We point out that in order to capture the chaotic dynamics that appear in some regions of the μ parameter (see Figures 1 and 2), the resolution of μ is increased by 10 times respect to the one of ν. We summarize the data generation procedure in Algorithm 1.

As described in Algorithm 1, for every pair of (μ, ν) values, we generate five trajectories or random length (between 10 and 50), each one with a random initial condition chosen in each one of these five ranges [0.0, 0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8], and (0.8, 1.0]. We also have set some restrictions on the admissible values in a trajectory: No value in the trajectory can be smaller than −1 nor greater than 3. If one of these conditions is satisfied, we remove the last point added to the trajectory and save the trajectory provided that it has a length greater than 9. Once a trajectory is generated, it is randomly assigned to the train, validation, or test data sets, resulting in a train-validation-test split with 1,110,266 trajectories in the train data set (65%), 255,667 trajectories in the validation data set (65%), and 341,374 in the test data set (20%).

Since the first layer in our neural network architecture is a convolutional layer, all trajectories have to be padded (if needed) with 0s to reach the maximum length (50). In other words, 0s are inserted to the left of the trajectory to make all of them have a length equal to 50, see [48, Ch. 5 and Ch. 9]. We also set a patience value equal to 20 to allow the training to end before the selected maximum number of epochs (200) if the training process does not improve the validation mean average error (MAE) after 20 consecutive epochs, thus saving training time. Finally, and for the sake of completeness, we

have used a computer with 16 cores configured with 128 GB RAM and Nvidia RTX 3090 GPU with 22 GB RAM, running Ubuntu 20.10. The complete training process took less than 3 h, running up to 24 epochs.

Algorithm 1 Data sets creation algorithm

```
for <every μ> do
    for <every ν> do
        <i = 0>
        while i < 5 do
            <Select random index>
            <Select random x(0)>
            <Select random trajectory length between 10 and 50
            <Generate trajectory>
            if trajectory length > 9 then
                if index ≤ 0.20 then
                    <Save test trajectory>
                else
                    if index ≤ 0.35 then
                        <Save validation trajectory>
                    else
                        <Save train trajectory>
                    end if
                end if
            else
                <pass>
            end if
            <i += 1>
        end while
    end for
end for
```

3 | RESULTS

In Figure 4, we depict the MAE distribution for μ and ν, comparing the ground truth (x-axis) and the predicted values (y-axis). The more values we have in the diagonal, the better the models can infer these values. The inference of the μ and ν parameters is very consistent along the entire range of possible values of each parameter. However, the MAE results for the inference of ν are slightly worse than the results for μ. We also notice that the trajectory length affects the accuracy of the predictions. Since we are dealing with time series and recurrent neural networks, it is expected that long trajectories will return lower MAE values.

In Table 1, we show the MAE results for different lengths bins. MAEs both for μ and ν are in the same order of magnitude (10^{-2}). This is the order taken for the ν discretization. This also supports the use of a thinner discretization for μ (10^{-3}) with respect to the one used for ν.

MAEs for short trajectories are in fact considerably larger than for the rest of lengths, for instance, more than 20,000 for length equal to 10 but around 6000 for lengths between 40 and 50. As it was also noticed in Conejero [41], the MAE slightly increases for trajectories with lengths between 40 and 50. It is worth to mention, that as the length increases, the number of trajectories in each bin decreases due to the bounds set for stopping the generation of trajectories (no term can be between smaller than -1 nor greater than 3).

In order to get deeper insights into the predictions obtained for the parameter ν, we explore in more detail the ones that result in high values of MAEs. In Figure 5, we represent the frequency of trajectories with MAEs higher than 0.05 (left) in terms of the initial conditions $x(0)$, trajectory length, and the μ and ν parameters. We appreciate than in most cases, the main causes are short trajectory lengths, values of μ near to 0 and an initial condition $x(0)$ close to 0.0 or to 1.0. On Figure 5 (right), we show the same histograms but just for trajectories of length greater than 15. Here, we see that despite

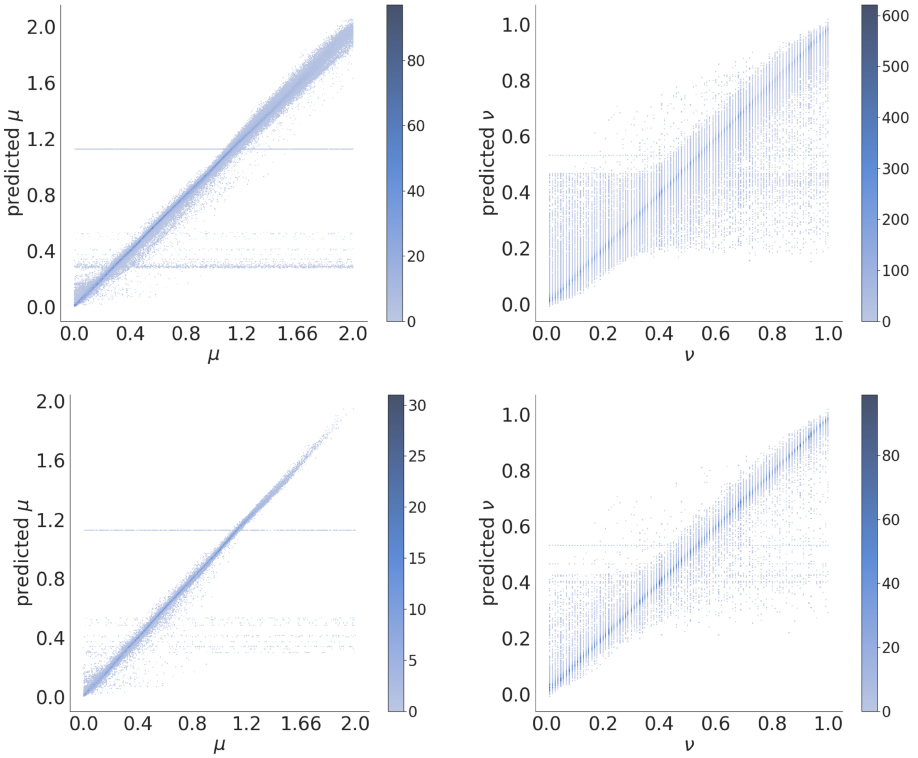

FIGURE 4 Truth versus predicted values of μ and ν in the validation data set (top) and for trajectories of length between 40 and 50 (bottom). [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 1 μ and ν MAEs in the test data set.

Length	MAE (μ)	MAE (ν)
10–19	0.0284	0.0318
20–29	0.0209	0.0229
30–39	0.0188	0.0217
40–50	0.0216	0.0227
All	0.0233	0.0259

the trajectory length disappears, we still appreciate the same conclusions for the values of $x(0)$ and μ. Besides, we also notice that whereas trajectories with values of μ between 0.4 and 1.1 return low MAEs for ν, values of μ close to 0 result in high MAE values for ν.

We also confront the values of the initial conditions with the values of μ and ν. In Figure 6, we represent the average MAE for pairs of initial conditions and values of the μ (left) and ν (right) parameters. The dark horizontal lines represent initial conditions that provide systematically higher values of MAE. On the one hand, for predictions of μ, it seems that the initial condition is much more relevant for values of μ smaller than 1.66 than for greater ones. On the other hand, for the predictions of ν, we can find initial conditions returning high values of MAE, mainly for values of ν smaller than 0.4.

We also study the MAE distribution for every μ and ν truth value. In Figure 7, we show the Q_1, Q_2, and Q_3 quartiles of these error distributions for each value of μ (left) and ν (right). On the one hand, it can be clearly seen that MAE error has greater variance for μ than for ν, in particular for values of μ greater than 1.25, which is consistent with what we observed in Figure 6. On the other hand, MAEs are considerably higher for ν rather than for μ, as we have previously deduced from the Feigenbaum diagrams in Figures 1 and 2, and from the comparison of predicted and truth values of μ and ν in Figure 4.

Due to the fractional nature of Equation (2), the model has a memory component that is strongly dependent on the initial condition $x(0)$. In Figure 8, we show boxplots of the MAE distribution for each initial condition $x(0)$. The initial condition is more influential in predicting μ rather than in predicting ν, since boxes (green) and whiskers (gray) are wider if we compare with the corresponding ones for ν. Again, we see that the results are worse for initial conditions close to 0 and to 1, due in part to the form of the nonlinear logistic terms in (2).

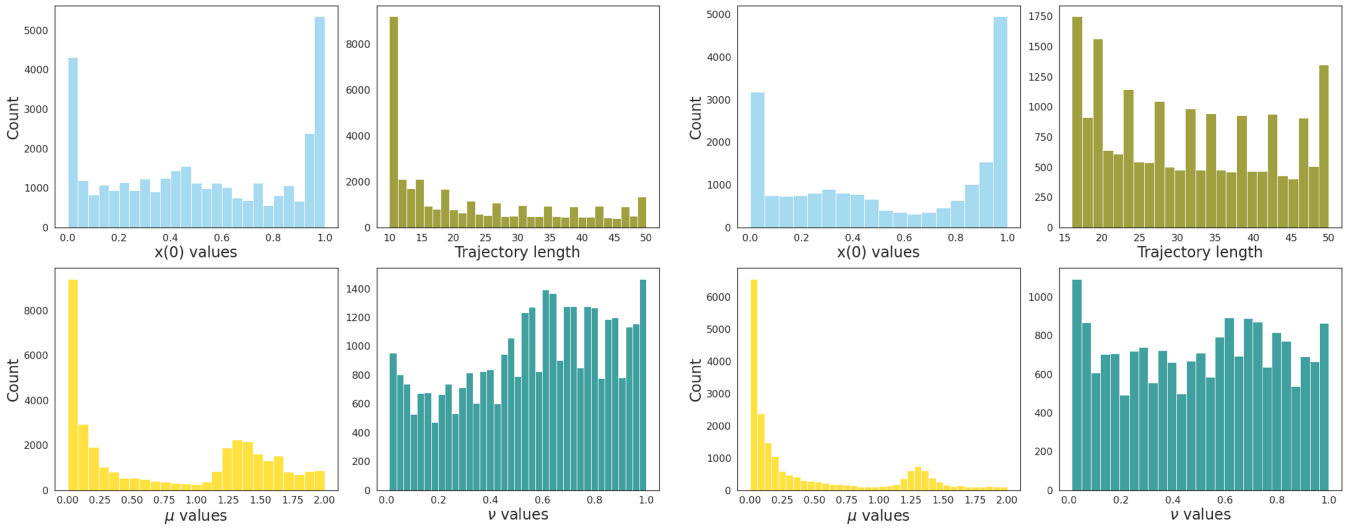

FIGURE 5 Histograms of $x(0)$, trajectory lengths, μ, and ν values for trajectories with ν MAE higher than 0.05 (left) and for trajectories with ν MAE higher than 0.05 and length greater than 15 (right). [Colour figure can be viewed at wileyonlinelibrary.com]

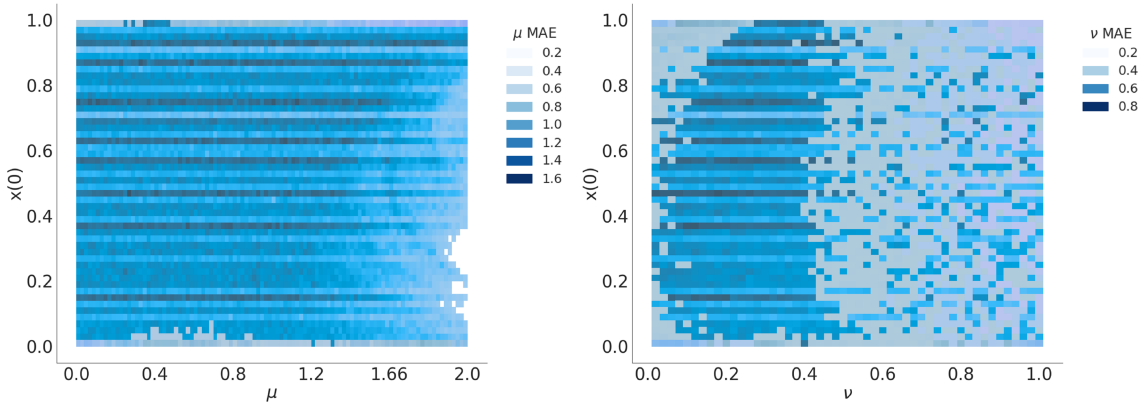

FIGURE 6 μ (left) and ν (right) values versus $x(0)$ by μ and ν MAE, respectively. [Colour figure can be viewed at wileyonlinelibrary.com]

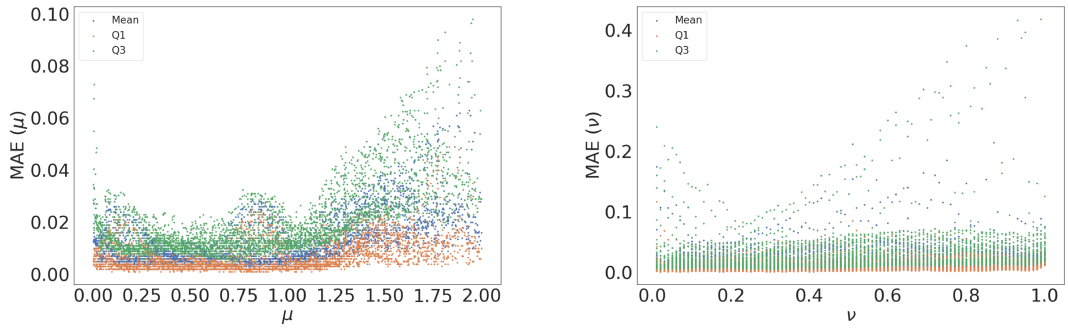

FIGURE 7 Quartiles Q1 (red), Q2 (blue), and Q3 (green) of the MAE distribution on the evaluation data for μ (left) and ν (right). [Colour figure can be viewed at wileyonlinelibrary.com]

Finally, we wonder if machine learning methods are able to determine which trajectories are generated from (1) and which ones are given by (2). In other words, we want to see if they can determine if a delayed is incorporated to the model or not. To do so, we have used the data set described in this work jointly with the data set used in Conejero et al. [41]. This second data set was also generated following the same procedure described in Algorithm 1 but with the following specifications.

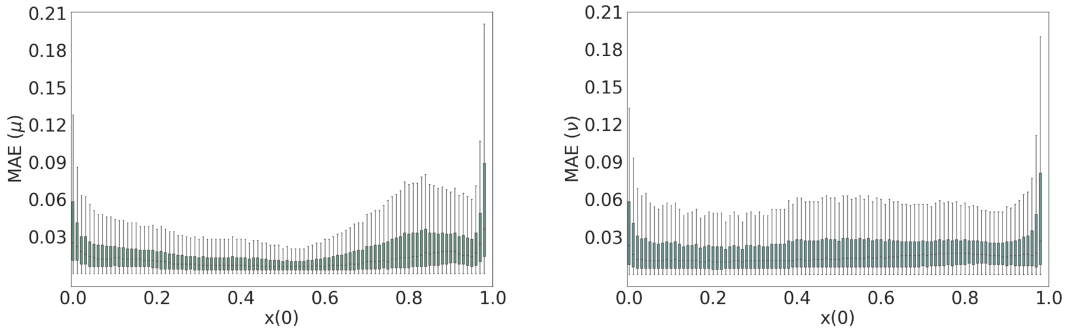

FIGURE 8 Box and whisker plots for the MAE distribution obtained from predictions of μ (left) and ν (right) on the evaluation data set in terms of the initial condition $x(0)$. The boxes are painted in green and the whiskers on gray. [Colour figure can be viewed at wileyonlinelibrary.com]

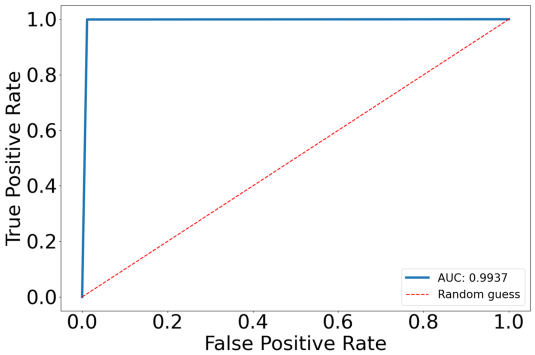

FIGURE 9 Receiver operating characteristic (ROC) curve on the test data set for the classification problem of fractional logistic trajectories with and without delay. [Colour figure can be viewed at wileyonlinelibrary.com]

- $\mu \in [2, 3.2]$ discretized with a step size of 10^{-3},
- $\nu \in [0.01, 1]$ discretized with a step size of 10^{-2},
- trajectory lengths between 10 and 50, and
- initial conditions $x(0) \in [0, 1]$ randomly chosen with a resolution of 10^{-2}.

Since the range of values for μ is bigger in the present work than in the other, we have randomly sampled the train, validation and test data sets in order to obtain three perfectly balanced pairs of sets, consisting of 618,199 trajectories for each training data set, 142,800 for each validation data set, and 190,334 for each the testing data set. For this classification task, we have used the architecture shown in Figure 3 but changing the activation function in the last dense layer to a sigmoid function. The training finishes after 43 epochs and 3 h and 30 min because the 20 epochs of patience are reached. The accuracy in the validation data set is of a 99.38%, which is very close to the area under the curve (AUC) value obtained for the test data set of 99.375%; see Figure 9 where we show the receiver operating characteristic (ROC) curve and the AUC value.

4 | CONCLUSIONS

In this work, we have considered a delayed version of the fractional logistic equation introduced by Wu and Baleanu [20]. We have seen that a combination of convolutional and recurrent neural networks succeeds in training a model that given a new trajectory produced by this model is able to infer the μ and ν parameters of the model. We have seen that the MAE of the predictions falls almost in the order of magnitude of the discretization used. It would be interesting to check up to which point the results can be improved using other deep learning models, training with larger data sets, and with data sets of longer trajectories.

In our results, we have seen the importance of the initial condition, as the main contributor to the memory effect of the generating model. We have seen that the initial conditions near 0 and 1 tend to accumulate predictions with higher errors

in comparison with other values. However, we have detected some bins of initial conditions that produce higher errors than other neighbor bins, and this happens almost independently of the values of the parameters μ and ν; see Figure 6. We have also seen that the predictor of the parameters provides pretty accurate predictions for trajectory lengths greater than 15 and, in general, the accuracy of the predictions increases as long as the trajectory length increases.

Finally, we have seen that these methods almost perfectly detect if a trajectory comes from a particular model with or without delay. It would be interesting to study whether this approach is able to determine the effect of the delayed term and of the fractional component in other fractional dynamical systems.

ACKNOWLEDGEMENTS

JAC acknowledges funding from Grant PID2021-124618NB-C21 funded by Ministerio de Ciencia e Innovación—Agencia Estatal de Investigación (MCIN/AEI/10.13039/501100011033) and by "ERDF A way of making Europe," by the European Union.

CONFLICT OF INTEREST STATEMENT

This work does not have any conflicts of interest.

ORCID

J. Alberto Conejero https://orcid.org/0000-0003-3681-7533
Òscar Garibo-i-Orts https://orcid.org/0000-0001-8089-1904
Carlos Lizama https://orcid.org/0000-0002-9807-1100

REFERENCES

1. G. C. Wu and D. Baleanu, *Discrete fractional logistic map and its chaos*, Nonlinear Dynam. **75** (2014), no. 1-2, 283–287.
2. L.-L. Huang, J. H. Park, G.-C. Wu, and Z.-W. Mo, *Variable-order fractional discrete-time recurrent neural networks*, J. Comput. Appl. Math. **370** (2020), 112633.
3. L.-L. Huang, G.-C. Wu, D. Baleanu, and H.-Y. Wang, *Discrete fractional calculus for interval–valued systems*, Fuzzy Sets Syst. **404** (2021), 141–158.
4. Y.-R. Bai, D. Baleanu, and G.-C. Wu, *A novel shuffling technique based on fractional chaotic maps*, Optik **168** (2018), 553–562.
5. F. Liu, M. Meerschaert, R. McGough, P. Zhuang, and Q. Liu, *Numerical methods for solving the multi-term time-fractional wave-diffusion equation*, Fract. Calc. Appl. Anal. **16** (2013), no. 1, 9–25.
6. G.-C. Wu, D. Baleanu, and Z.-X. Lin, *Image encryption technique based on fractional chaotic time series*, J. Vib. Control **22** (2016), no. 8, 2092–2099.
7. G.-C. Wu, Z.-G. Deng, D. Baleanu, and D.-Q. Zeng, *New variable-order fractional chaotic systems for fast image encryption*, Chaos **29** (2019), no. 8, 83103.
8. L. Cristofaro, R. Garra, E. Scalas, and I. Spassiani, *A fractional approach to study the pure-temporal Epidemic Type Aftershock Sequence (ETAS), process for earthquakes modeling*, 2022. arXiv preprint arXiv:2301.02586.
9. H. Kong, G. Yang, and C. Luo, *Modeling aftershocks by fractional calculus: exact discretization versus approximate discretization*, Fractals **29** (2021), no. 8, 2140038.
10. M. Ye, J. Li, X. Han, and H. Jiang, *Global analysis of a fractional-order reaction-diffusion rumor-spreading model in multi-lingual environment*, Eur. Phys. J. Plus **138** (2023), no. 1, 102.
11. B. Kuttner, *On differences of fractional order*, Proc. London Math. Soc. **3** (1957), 453–466.
12. J. B. Diaz and T. J. Osler, *Differences of fractional order*, Math. Comp. **28** (1974), 185–202.
13. H. L. Gray and N. F. Zhang, *On a new definition of the fractional difference*, Math. Comp. **50** (1988), 513–529.
14. K. S. Miller and B. Ross, Fractional difference calculus, *Univalent functions, fractional calculus, and their applications (Köriyama, 1988)*, Horwood, Chichester, 1989, pp. 139–152.
15. F. M. Atici and P. W. Eloe, *A transform method in discrete fractional calculus*, Int. J. Differ. Equ. **2** (2007), 165–176.
16. G.-C. Wu, D. Baleanu, and S.-D. Zeng, *Discrete chaos in fractional sine and standard maps*, Phys. Lett. A **378** (2014), no. 5-6, 484–487.
17. J. A. Conejero, C. Lizama, A. Mira-Iglesias, and C. Rodero, *Visibility graphs of fractional Wu–Baleanu time series*, J. Differ. Equ. Appl. **25** (2019), no. 9-10, 1321–1331.
18. T. T. Song, G.-C. Wu, and J.-L. Wei, *Hadamard fractional calculus on time scales*, Fractals **30** (2022), no. 7, 1–14.
19. G. C. Wu, T. T. Song, and S. Q. Wang, *Caputo-Hadamard fractional differential equations on time scales: numerical scheme, asymptotic stability, and chaos*, Chaos **32** (2022), no. 9, 93143.
20. G. C. Wu and D. Baleanu, *Discrete chaos in fractional delayed logistic maps*, Nonlinear Dynam. **80** (2015), no. 4, 1697–1703.
21. G. Muñoz-Gil, G. Volpe, M. A. García-March, R. Metzler, M. Lewenstein, and C. Manzo, *The anomalous diffusion challenge: Objective*

comparison of methods to decode anomalous diffusion, Emerging Topics in Artificial Intelligence (ETAI) 2021 *Edited by* G. Volpe, J. B. Pereira, D. Brunner, and A. Ozcan, Vol. **11804**, San Diego, California, International Society for Optics and Photonics, SPIE, 2021, pp. 1180416.
22. P. Massignan, C. Manzo, J. A. Torreño-Pina, M. F. Garcia-Parajo, M. Lewenstein, and G. J. Lapeyre, *Nonergodic subdiffusion from Brownian motion in an inhomogeneous medium*, Phys. Rev. Lett. **112** (2014), no. 15, 150603.
23. H. Scher and E. W. Montroll, *Anomalous transit-time dispersion in amorphous solids*, Phys. Rev. B **12** (1975), 2455–2477.
24. J. H. Jeon and R. Metzler, *Fractional Brownian motion and motion governed by the fractional Langevin equation in confined geometries*, Phys. Rev. E **81** (2010), 21103.
25. B. B. Mandelbrot and J. W. Van Ness, *Fractional Brownian motions, fractional noises and applications*, SIAM Rev. **10** (1968), no. 4, 422–437.
26. J. Klafter and G. Zumofen, *Lévy statistics in a Hamiltonian system*, Phys. Rev. E **49** (1994), 4873–4877.
27. G. Muñoz-Gil, G. Volpe, M. A. Garcia-March, E. Aghion, A. Argun, C. B. Hong, T. Bland, S. Bo, J. A. Conejero, N. Firbas, and Ò. Garibo-I Orts, *Objective comparison of methods to decode anomalous diffusion*, Nature Commun. **12** (2021), no. 1, 6253.
28. S. C. Lim and S. V. Muniandy, *Self-similar Gaussian processes for modeling anomalous diffusion*, Phys. Rev. E **66** (2002), no. 2 Pt 1, 21114.
29. A. Argun, G. Volpe, and S. Bo, *Classification, inference and segmentation of anomalous diffusion with recurrent neural networks*, J. Phys. A: Math. Theor. **54** (2021), no. 29, 294003.
30. Ò. Garibo-I Orts, A. Baeza-Bosca, M. A. Garcia-March, and J. A. Conejero, *Efficient recurrent neural network methods for anomalously diffusing single particle short and noisy trajectories*, J. Phys. A: Math. Theor. **54** (2021), no. 50, 504002.
31. A. Gentili and G. Volpe, *Characterization of anomalous diffusion classical statistics powered by deep learning (CONDOR)*, J. Phys. A: Math. Theor. **54** (2021), no. 31, 314003.
32. M. A. Z. Raja, M. Umar, Z. Sabir, J. A. Khan, and D. Baleanu, *A new stochastic computing paradigm for the dynamics of nonlinear singular heat conduction model of the human head*, Eur. Phys. J. Plus **133** (2018), no. 9, 1–21.
33. Z. Sabir, M. A. Z. Raja, D. Baleanu, K. Cengiz, and M. Shoaib, *Design of Gudermannian Neuroswarming to solve the singular Emden–Fowler nonlinear model numerically*, Nonlinear Dynam. **106** (2021), no. 4, 3199–3214.
34. A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F. J. R. Ruiz, J. Schrittwieser, G. Swirszcz, and D. Silver, *Discovering faster matrix multiplication algorithms with reinforcement learning*, Nature **610** (2022), no. 7930, 47–53.
35. D. Baleanu, Z. Güvenç, J. A. Burhanettin, and T. Machado, *New trends in nanotechnology and fractional calculus applications*, Vol. **10**, Springer, 2010.
36. J. A. Conejero, J. Franceschi, and E. Picó-Marco, *Fractional vs. ordinary control systems: what does the fractional derivative provide?* Mathematics **10** (2022), no. 15, 2719.
37. E. İlhan and İ. O. Kıymaz, *A generalization of truncated M-fractional derivative and applications to fractional differential equations*, Appl. Math. Nonlinear Sci. **5** (2020), no. 1, 171–188.
38. H. G. Sun, Y. Zhang, D. Baleanu, W. Chen, and Y. Q. Chen, *A new collection of real world applications of fractional calculus in science and engineering*, Commun. Nonlinear Sci. Numer. Simulat. **64** (2018), 213–231.
39. D. Valério, M. D. Ortigueira, and A. M. Lopes, *How many fractional derivatives are there?* Mathematics **10** (2022), no. 5, 737.
40. M. Ortigueira and J. A. T. Machado, *Which derivative?* Fract. Fractional **1** (2017), no. 1, 3.
41. J. A. Conejero, Ó. Garibo-i Orts, and C. Lizama, *Inferring the fractional nature of Wu Baleanu trajectories*, Research Square 2022. DOI 10.21203/rs.3.rs-2218679/v1
42. M. A. Lozano, Ò. Garibo-I Orts, E. Piñol, M. Rebollo, K. Polotskaya, M. A. Garcia-March, J. A. Conejero, F. Escolano, and N. Oliver, *Open data science to fight COVID-19: winning the 500k XPRIZE pandemic response challenge*, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Bilbao, Spain, Springer, 2021, pp. 384–399.
43. A. Krizhevsky, V. Nair, and G. Hinton, *CIFAR-10 (Canadian Institute for Advanced Research)* **5** (2010), no. 4, 1. URL http://www.cs.toronto.edu/kriz/cifar.html
44. A. Krizhevsky, I. Sutskever, and G. E. Hinton, *ImageNet classification with deep convolutional neural networks*, Proceedings of the 25th International Conference on Neural Information Processing Systems, NIPS'12, Vol. **1**, Curran Associates Inc, Red Hook, NY, USA, 2012, pp. 1097–1105.
45. S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural Comput. **9** (1997), no. 8, 1735–1780.
46. Z. C. Lipton, *A critical review of recurrent neural networks for sequence learning*, 2015. arXiv, abs/1506.00019.
47. D. P. Kingma and J. Ba, *Adam: a method for stochastic optimization*, 2014. arXiv preprint arXiv:1412.6980.
48. I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, Adaptive computation and machine learning, Boston (MA), USA, MIT Press, 2016.

How to cite this article: J. A. Conejero, Ò. Garibo-i-Orts, and C. Lizama. *Recovering discrete delayed fractional equations from trajectories*, Math. Meth. Appl. Sci. (2023), 1–10. DOI 10.1002/mma.9228