Chinese Journal of Physics 89 (2024) 1276–1285

Contents lists available at ScienceDirect

Chinese Journal of Physics

journal homepage: www.elsevier.com/locate/cjph

Predictive deep learning models for analyzing discrete fractional dynamics from noisy and incomplete data

Òscar Garibo-i-Orts [a,b], Carlos Lizama [c], Ali Akgül [d,e], J. Alberto Conejero [a,*]

[a] Instituto Universitario de Matemática Pura y Aplicada Universitat Politècnica de València, Spain
[b] GRID - Grupo de Investigación en Ciencia de Datos, Valencian International University - VIU, Carrer Pintor Sorolla 21, València, 46002, Spain
[c] Departamento de Matemática y Ciencia de la Computación, Universidad de Santiago de Chile, Las Sophoras 173, Estación Central, Santiago, Chile
[d] Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon
[e] Siirt University, Art and Science Faculty, Department of Mathematics, Siirt, 56100, Turkey

ARTICLE INFO

Keywords:
Dynamical systems
Discrete fractional calculus
Wu–Baleanu model
Logistic map
Convolutional neural networks
LSTM networks

ABSTRACT

We study the accuracy of machine learning methods for inferring the parameters of noisy fractional Wu–Baleanu trajectories with some missing initial terms. Our model is based on a combination of convolutional and recurrent neural networks (LSTM), which permits the extraction of characteristics from trajectories while preserving time dependency. We show that these approach exhibit good accuracy results despite the poor quality of the data.

1. Introduction

Discrete Fractional Calculus (DFC) is a valuable tool to characterize dynamical systems representing real-world processes of discrete nature with long-term connections between different time steps. In particular, DFC is a powerful tool for modeling phenomena in science and engineering that exhibit non-local and memory properties. Since the first preliminary works on discrete fractional calculus [1–4], it has attracted the interest of researchers in the analysis of dynamical systems [5–8], with the recent appearance of new definitions of fractional derivatives [9].

Until 1976, when May [10] introduced the logistic equation (1), dynamical properties of systems were described using simple first-order difference equations, which implied that nonlinear dynamical characteristics were not considered. The logistic equation is expressed as:

$$x(n+1) = \eta x(n)(1 - x(n)), \qquad \text{for } n \in \mathbb{N}_0, \tag{1}$$

where $x(0) \in [0, 1]$ and $\eta \in \mathbb{R}$. This equation is well defined in the interval $0 \le \eta \le 4$, converging to 0 if $0 \le \eta < 1$, and presenting nontrivial dynamical behavior for $1 \le \eta \le 4$. Applying the change of variable $x(n) = \frac{\eta}{\eta-1} x(n)$, one can transform the logistic equation to extend it with a discrete fractional derivative. Therefore, alternately to compute the term $x(n+1)$ by recurrence, the forward Euler operator Δ can substitute the nonlinear right term of the logistic as follows

$$\Delta x(n) := x(n+1) - x(n), \tag{2}$$

* Corresponding author.
E-mail address: aconejero@upv.es (J.A. Conejero).

https://doi.org/10.1016/j.cjph.2024.04.010
Received 11 November 2023; Received in revised form 5 April 2024; Accepted 9 April 2024
Available online 12 April 2024

having added the difference operator, which allows the derivative approximation, rescaling the initial condition by a factor $\frac{\mu+1}{\mu}$, yielding

$$\Delta x(n) = \mu x(n)(1 - x(n)), \qquad x(0) = \frac{\mu + 1}{\mu} x(0). \tag{3}$$

A natural generalization to discrete fractional calculus of the logistic equation through a discrete fractional operator and the Caputo derivative was provided by [11]. They started from the expression $\Delta x(n) := x(n + 1) - x(n)$, where Δ is the forward Euler operator, and replaced the Euler operator with the left Caputo discrete difference operator Δ^{ν} as follows:

$$\Delta^{\nu} x(n) := \sum_{j=0}^{n} k^{-\nu}(n - j)x(j), \quad n \in \mathbb{N}_0, \tag{4}$$

where $k^{-\nu}(j)$ is defined by the generating series

$$\sum_{j=0}^{\infty} k^{-\nu}(j)z^j = (1 - z)^{\nu}, \quad \nu \in \mathbb{R}, \tag{5}$$

which in turn results in the following expression for the fractional logistic map

$$x(n) = x(0) + \frac{\mu}{\Gamma(\nu)} \sum_{j=1}^{n} \frac{\Gamma(n - j + \nu)}{\Gamma(n - j + 1)} x(j - 1)(1 - x(j - 1)), \tag{6}$$

where μ is a parameter, and ν indicates the fractional derivation order. This expression can also be stated as a convolution, through 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 fractional derivation order ν [12].

When trajectories are obtained from physical experiments, they may be noisy or incomplete. Under these conditions, machine learning techniques may help us to overcome these difficulties and help us to infer the model's parameters from such poor data. It is worth noting that while they have shown great promise in modeling complex systems, they also have limitations. For example, the accuracy of these methods heavily depends on the quality and quantity of available data, which can be a challenge in real-world applications.

Machine learning has provided powerful tools for estimating model parameters, as we can see in the case of anomalous diffusion [13–17], when inferring the anomalous diffusion exponent. This has been possible if we process trajectories with machine learning models based on LSTM recurrent neural networks [18–20] and transformers [21] and deep learning [22,23] architectures. The LSTM and transformers architectures are able to learn from sequential ordered data despite some values were missing or slightly modified. The values with more impact in the prediction are the ones in the last positions.

In fractional calculus, inference of missing parameters and model structure from fractional models has been achieved using evolutionary algorithms such as the composite differential evolution (CoDE) algorithm [24] or by refined nonlinear least-squares objective functions [25]. Machine learning methods have been recently used for inferring parameter models in discrete fractional models [26,27].

We have also seen that these models can help us infer the underlying model's parameters when we have short trajectories from a Wu–Baleanu model [28] or from the corresponding delayed model [29] obtaining on average, an error of the same order of magnitude as the parameter step size discretization. A more general panoramic of the last advances of the conjunction of fractional calculus and machine learning can be found in [30–34].

In trajectories with memory, as in the case of the fractional models, this can be extremely helpful when training and testing the model for obtaining a high accuracy. Therefore, we also want to explore how accurate these models can be when dealing with trajectories with some initial missing terms. Besides, since we want to know to which extent these models can help to deal with experimental data, we also have incorporated noise in the trajectories at each step in the trajectory, as this is a common practice when developing models to infer parameters from experimental data [13,35,36].

In this paper, we investigate the accuracy of machine learning methods for modeling Wu–Baleanu trajectories in the presence of noise and with missing initial terms. Our work is organized as follows: Section 2 describes how the data was obtained and the configuration of the machine learning model. Later, in Section 3, we show the results obtained when predicting the model's parameters. Finally, in Section 4, we summarize the results and discuss their limitation.

2. Methodology

The combination of convolutional and LSTM networks to work with time series has recently been used in the study of time series of many types, in particular for trajectories obtained from dynamical systems, either to predict the behavior of the system [37–40]. See also for its comparison with transformer-based architectures [41]. Sometimes, the mix of convolutional neural networks (CNNs) with LSTMs us to extract informative characteristics from the trajectory while preserving the temporal dependencies, improving the accuracy of the models [42–45]. Such an idea has been successfully applied in different applications such as in astrophysics [46,47], weather forecasting [48], or signal processing [49,50].

In this work, we use the architecture described in [28]: two convolutional layers, three bidirectional Long Short Term Memory (LSTM) networks [51], and two dense layers. Primarily, data flows through the convolutional layers, which behave as a feature extractor or encoder, and their output becomes the input for the LSTM layers, which learn the temporal patterns.

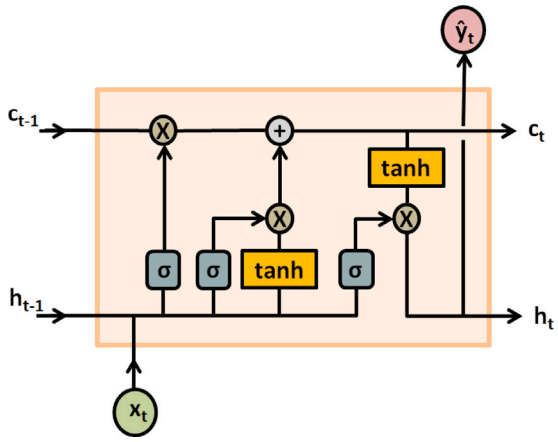

Fig. 1. Scheme of an LSTM layer.

The two convolutional layers are used for feature extraction. A one-dimensional (1D) convolutional layer can extract spatial features from a 1D trajectory. The kernel of size five behaves as a sliding window that moves along the trajectory, focusing on different parts of it and extracting features related to the spatial arrangement of the trajectory. Stacking two convolutional layers results in the extraction of characteristics of different levels of abstraction. The three bidirectional LSTM layers learn from the sequential features extracted by the convolutional networks. In Fig. 1, we depict a scheme of an LSTM unit; an LSTM layer comprises a given number of these units concatenated. LSTM units include a cell state (denoted by h, see Fig. 1) that is passed from time t to time $t + 1$ and allows to retain some memory of the critical information in a trajectory. In this way, the LSTM unit output combines both the input at time t and the inputs of the time steps prior to t, which are kept by the cell state h. Since the information kept or removed from the cell state is done in one direction, we use bidirectional LSTM, which simultaneously performs the same operation but forward and backward. Therefore, any critical information that the forward LSTM could have discarded could be incorporated by the backward one, and vice-versa.

We have trained four different models for simultaneously predicting μ and ν, that are described as follows:

1. *Raw model (R)*: This model has been trained with trajectories from the fractional version of the logistic equation by Wu and Baleanu as in (6). The trajectories were generated with the following ranges for the parameters: $\mu \in [2.0, 3.2]$ with increments of 0.001, $\nu \in [0.001, 1.0]$ with an increment of 0.01, and $x(0)$ (initial value) $\in [0.0, 1.0]$ with increments of 0.01.
2. *Noisy model (N)*: We have trained a model with noisy trajectories with the same range of parameters as in the previous case. We have added noise sampled from a Gaussian distribution with 0 mean and standard deviation equal to 0.01. Given $x(0)$ as the initial condition and ε_0 as a noise term, we define $\tilde{x}(0) = x(0) + \varepsilon_0$, where. We compute $x(1)$ as in (6) using $\tilde{x}(0)$ as initial condition. Then we add a noise term ε_1, and this new value will be denoted as $\tilde{x}(1) = x(1) + \varepsilon_1$. We proceed inductively and compute $\tilde{x}(n)$

$$\tilde{x}(n) = x(0) + \varepsilon_0 + \frac{\mu}{\Gamma(\nu)}\left(\sum_{j=1}^{n}\frac{\Gamma(n-j+\nu)}{\Gamma(n-j+1)}\tilde{x}(j-1)(1-\tilde{x}(j-1))\right) + \varepsilon_n, \tag{7}$$

   where ε_n is a noise term added in the time step n. Therefore, the noise in the trajectory is incremental.
3. *Raw missing data model (RM)*: This model has been trained with a data set built like in the raw model *R*. However, after computing each trajectory, we remove a random number between 5 and 10 of the initial terms from that trajectory.
4. *Noisy missing data model (NM)*: This model has been trained with a data set built with the same procedure as in the noisy model *N* case, but we remove a random number between 5 and 10, of the initial terms from each trajectory, as in the previous case.

For each model, we generate a training data set (learn the data distribution) and a validation data set to evaluate the model's performance at each epoch (or iteration over the data) during the training stage, enabling the model to adjust the internal weights after each epoch to improve the accuracy. We also generate four data sets for testing, one per model.

All trajectories have been generated with a variable length ranging from 10 to 50. The reason is that in many experiments, such as single-particle-tracking ones, it is very common that the lengths of trajectories obtained from experiments were really short [52]. This premise limits the number of initial elements that can be removed from any trajectory. Since data input to a convolutional layer needs to be of a fixed length, we insert as many zeros as needed at the beginning of each trajectory (in other words, we pad the trajectories) to make them of length 50. Therefore, all trajectories will be of the same length, even after removing initial elements, and we let the model learn that these initial elements with values equal to zero are of no importance to infer the μ and ν parameters. In Table 1, we show the data sets used to train, validate, and test the models.

Table 1
Number of trajectories in the train, validation, and test data sets generated per each model.

	Train	Validation	Evaluation
Raw model	618,199	142,800	190,334
Noisy model	612,624	141,435	187,566
Raw missing data model	465,349	107,563	143,050
Noisy missing data model	458,103	106,064	141,187

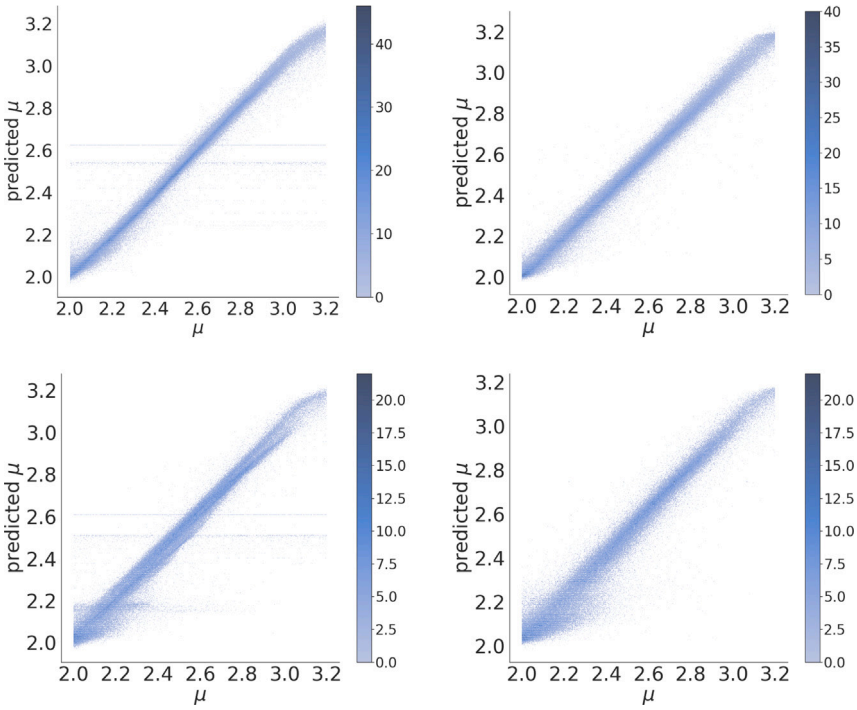

Fig. 2. μ predictions scatter plot for each model with its own test data: R on E_R (up left), N on E_N (up right), RM on E_{RM} (bottom left), and NM on E_{NM} (bottom right).

The aforementioned training data sets will be denoted as T_R, T_N, T_{RM}, and T_{NM} respectively, while V_R, V_N, V_{RM}, and V_{NM} denote the validation data sets, and the test data sets E_R, E_N, E_{RM}, and E_{NM} used for evaluation. We will refer to them to evaluate the performance on the prediction of μ and ν parameters.

Our R, N, RM, and NM models are used to predict the μ and ν parameters' values, and these predictions were evaluated with the actual values for μ and ν. In order to shed light on the importance of the addition of noise and the deletion of some of the initial terms of a trajectory, we will evaluate each model with its corresponding test data set and with all the other test data sets. In this way, we analyze how the elements removal, the addition of noise, or the combination of both actions affect the models' ability to infer the μ and ν parameters. Sixteen experiments were conducted, as every model was evaluated on every test data set. Our results to evaluate the accuracy when inferring the μ and ν parameters are expressed in terms of the mean absolute error (MAE), defined as the mean of the absolute value of the errors obtained when inferring all the parameters of the trajectories in a given data set. We analyze the obtained results in the next section.

3. Results

We discuss the performance of the four models over the aforementioned described models. Firstly, Fig. 2 shows a scatter plot of the real (x-axis) and predicted values (y-axis) of the parameter μ given by each model over its corresponding test data set. It is straightforward but worth mentioning that the closer the resulting values are to the diagonal, the better the model's performance at inferring the μ values is.

In Table 2, we show the MAE of each model when evaluated on each evaluation data set for predicting μ. We see that each model outperforms others when tested on its corresponding evaluation data set. Nevertheless, in all the cases, the MAE is within the same order of magnitude as the discretization used for generating the trajectories, in the line of what was already observed in [28].

Despite all the models used for inferring μ show very small errors, looking back at Fig. 2, models used over complete trajectories (R and N) show a homogeneous behavior along the entire range of values of μ. In contrast, the models RM and NM show higher error variability (more points separated from the diagonal) for lower values of μ.

Table 2
MAE results for the models R, N, RM, and NM evaluated on the data sets E_R, E_N, E_T and E_{TN} for predicting μ.

	E_R	E_N	E_{RM}	E_{NM}
Raw model (R)	**0.025329**	0.035202	0.141953	0.145223
Noisy model (N)	0.027816	**0.025865**	0.163869	0.163828
Raw missing data model (RM)	0.046035	0.075430	**0.04244**	0.077100
Noisy missing data model (NM)	0.046757	0.046734	0.046294	**0.043667**

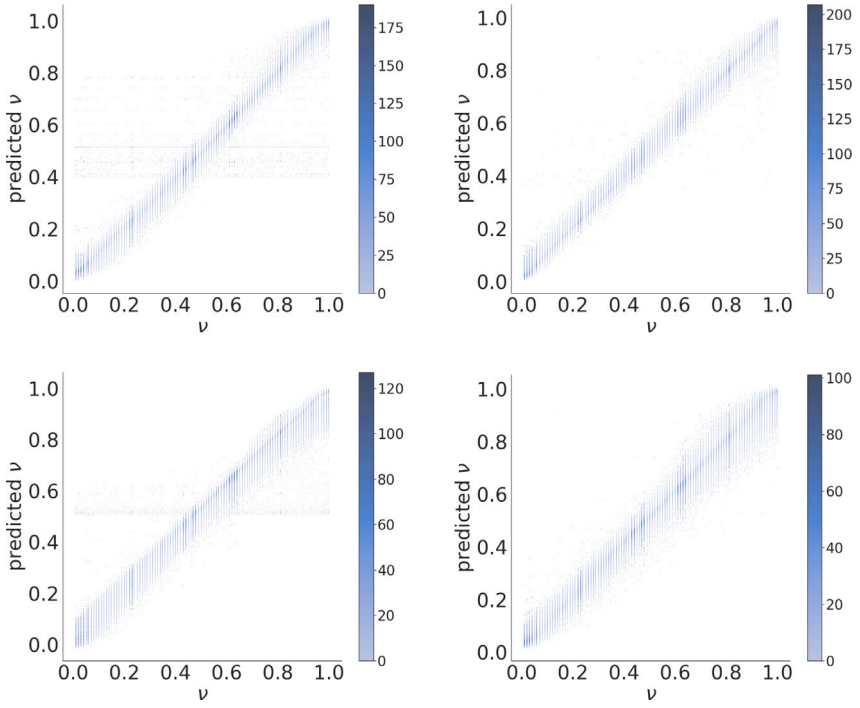

Fig. 3. Scatter plots for the predictions of ν from each model evaluated on its own evaluation data: R on E_R (up left), N on E_N (up right), RM on E_{RM} (bottom left), and NM on E_{NM} (bottom right).

Table 3
MAE results for the models R, N, RM, and NM evaluated on the data sets E_R, E_N, E_{RM} and E_{NM} for predicting ν.

	E_R	E_N	E_{RM}	E_{NM}
Raw model	**0.018583**	0.021626	0.058359	0.058972
Noisy model	0.021908	**0.017560**	0.087582	0.084969
Raw missing data model	0.033385	0.040394	0.032015	0.041381
Noisy missing data model	0.029028	0.026269	**0.0287648**	**0.024460**

Similarly, Fig. 3 shows the scatter plot of the predictions of ν by each considered model with respect to their corresponding evaluation data sets: R respect to $E_R(\nu)$, N respect to E_N and so on. Here, MAEs are lower than the ones obtained for μ. We also see that the predictions present higher variability for values of ν close to 0 or to 1 than in the rest of the range of ν.

Table 3 shows the MAEs obtained by each model when evaluated on the four evaluation data sets. Again, for each data set, its corresponding model outperforms the other ones, except for the case of E_{RM} in which the other model generated with missing values, NM, presents better results than the RM model.

In Fig. 4, we can quickly notice that adding incremental noise to the trajectories worsens the model's accuracy as expected. However, we can still infer the μ value within an acceptable error margin along the complete range of values. Nevertheless, removing some steps from the beginning of the trajectories results in much more error when inferring the μ values, which is coherent with the results presented in [28,29], where the importance of the starting point to generate the trajectories to determine its behavior was established.

An analogous result can be seen when analyzing the performance of the noisy model N. As shown in Fig. 5, the model N trained with cumulative noise in the trajectories presents a similar performance as the raw model R on the evaluation data sets E_R and E_N. However, the lack of the initial terms of the trajectory highly affects the model's performance. The horizontal spot of dots at height

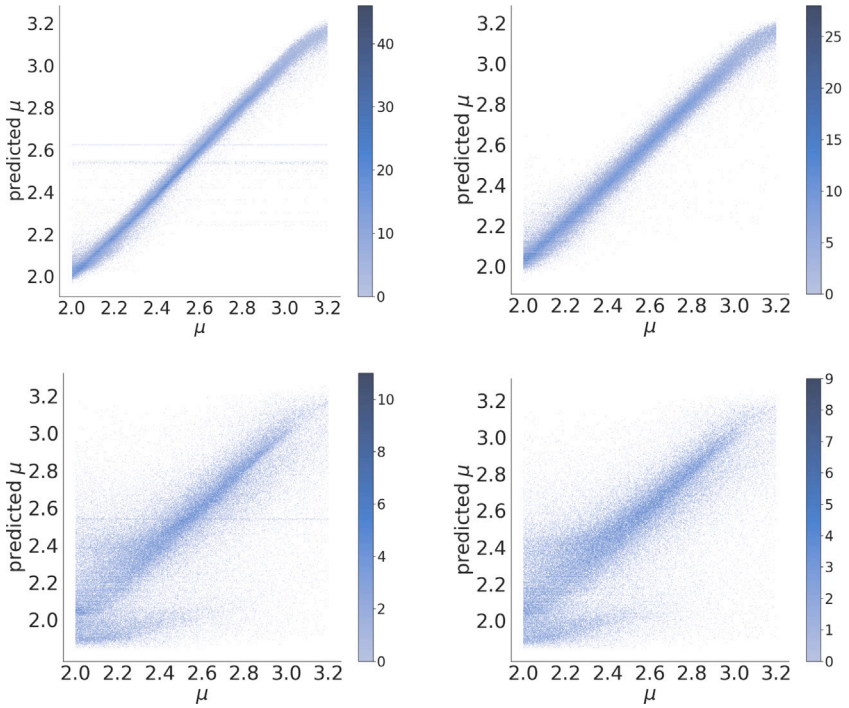

Fig. 4. Scatter plots for the predictions of μ obtained from the raw model R evaluated on the data sets E_R (up left), E_N (up right), E_{RM} (bottom left), and E_{NM} (bottom right).

2 on the scatter plots of model N evaluated on E_{RM} and E_{NM} show how the model has difficulties when predicting the values of ν.

In Fig. 6, we show the results of the RM model evaluated on the different evaluation data sets for predicting μ. As seen in Table 2, this model achieves its best performance on the E_R and E_{RM} data sets. Adding incremental noise introduces errors in the model predictions, especially for values of μ in the range $[2, 2.5]$.

In Fig. 7, we depict the results for the NM model. Interestingly, predictions tend to fall below the diagonal in contrast to what can be seen in Fig. 6. However, in these cases, the error made when predicting the μ value is more significant, with no noticeable influence of the absence or presence of cumulative noise in the test data sets E_N and E_{NM}.

Similar results can be seen with the parameter ν inference. In Fig. 8, we depict the performance of all four models R, N, RM, and NM when predicting the parameter ν on E_{NM}. We can see that the model trained with missing initial terms achieves better performance than the others, outstanding with the E_{NM} data set. That is, with data similar to the used to train the model. Here, it is noticeable how the initial conditions, or starting point, are conditioning the model's performance, as we pointed out when inferring the μ parameter.

We have also compared the accuracy of these models in the inference of the parameters when the missing values are randomly distributed along the sequence and not extracted from the beginning of the trajectory.

We have generated new data sets of equal size for training validation and evaluation of new raw models. On the one hand, when predicting μ, the new raw model has a MAE of 0.056875 compared to 0.04244. On the other hand, when predicting ν, the model provides a MAE of 0.043843, which is higher than the MAE provided by the raw model trained with trajectories with missing values at the beginning of the trajectory, with a MAE of 0.032015.

To see the importance of removing missing values at the beginning of the trajectory, we have tried another experiment. We have chose a random position in the trajectory and we have replaced a consecutive number of random positions, between 5 and 10.

The errors are still higher respect to the raw model, with MAE of 0.047577 instead of 0.04244 when predicting μ and MAE of 0.037227 instead of 0.032015 when predicting ν. This also shows that removing trajectories at the beginning of the trajectory is less relevant than removing in other positions which is linked to the LSTM architecture that gives more importance to the last elements than to the first ones.

4. Conclusions

In this work, we have analyzed the usefulness of machine learning methods to infer the parameters of fractional Wu–Baleanu trajectories under four scenarios simulating data obtained from physical experiments, like noisy or incomplete trajectories.

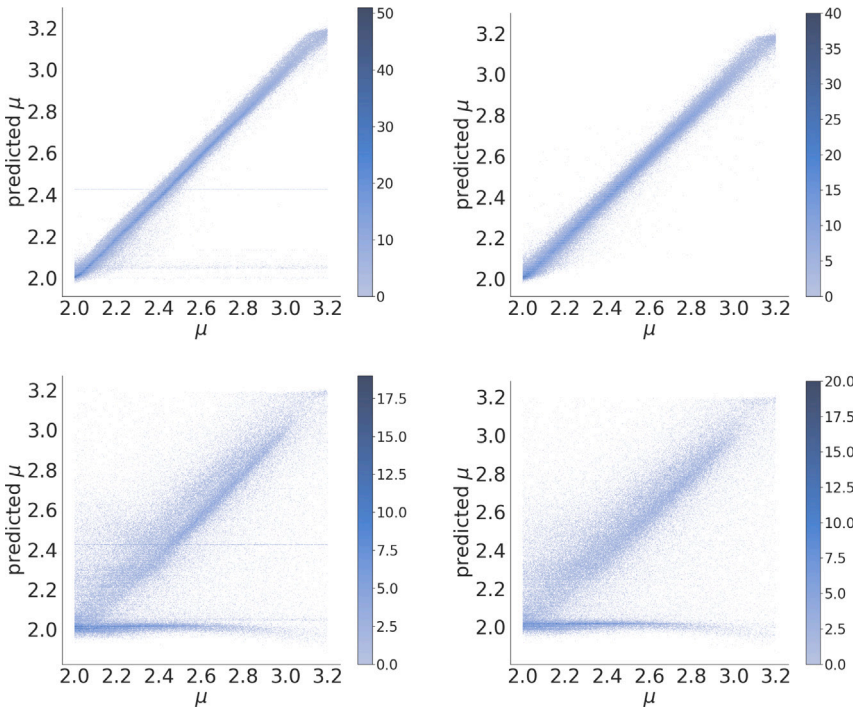

Fig. 5. Scatter plots for the predictions of μ obtained from the noisy model N evaluated on the data sets E_R (up left), E_N (up right), E_{RM} (bottom left), and E_{NM} (bottom right).

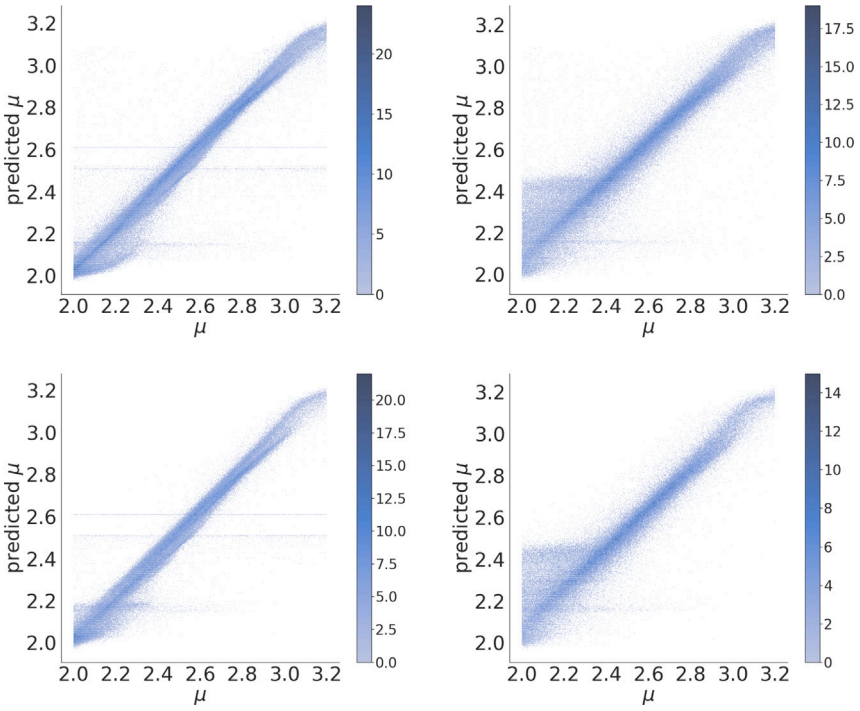

Fig. 6. Scatter plots for the predictions of μ obtained from the raw model trained with missing initial terms RM evaluated on the data sets E_R (up left), E_N (up right), E_{RM} (bottom left), and E_{NM} (bottom right).

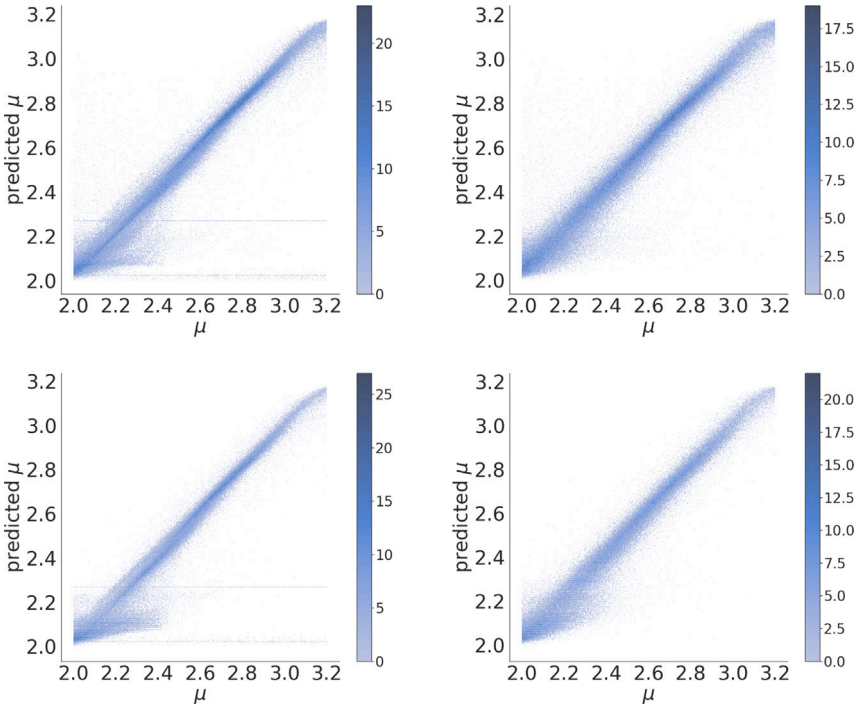

Fig. 7. Scatter plots for the predictions of μ obtained from the raw model trained with missing initial terms NM evaluated on the data sets E_R (up left), E_N (up right), E_{RM} (bottom left), and E_{NM} (bottom right).

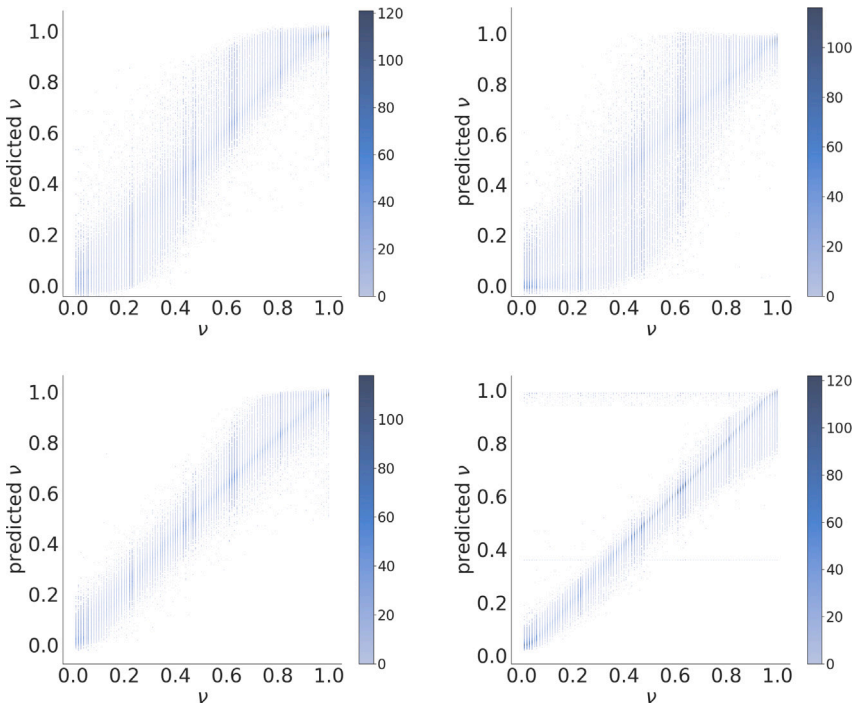

Fig. 8. Scatter plots for the predictions of ν obtained from the four models, raw model R (up left), noisy model N (up right), raw missing data model RM (bottom left), and noisy missing data model NM (bottom right), evaluated on the E_{NM} data set.

We have shown that models trained with data corresponding to each scenario can effectively infer the μ and ν parameters. Additionally, we have pointed out the importance of having complete trajectories, in the sense of what we pointed out in our work [28] since models trained with raw or noisy data struggle to correctly predict both parameters, while models trained with truncated data (noisy or not) obtain more accurate predictions for both parameters. Moreover, models trained with truncated trajectories can, on average, predict μ and ν parameters with lower error, suggesting that despite the error being higher than when using the corresponding models, the models can still generalize the distribution of the trajectories. We have also seen that if the initial terms are missing, the errors are, in general pretty smaller than when removing items from other positions in the sequence.

Even in the presence of chaos, the models do not show a worse accuracy, as it can be deduce when comparing the Feigenbaum diagrams from [28] where, depending on the values ν, chaos starts to appear beyond 2.2 or 2.3, but not close to 0.

Given a trajectory susceptible to showing fractional behavior, the four models used in this paper can be used to infer the μ and ν parameters, which can be used to generate a new trajectory to be compared to the original one and find out the parameters' values that fit the original trajectory the best. We also suggest studying models for classifying a given trajectory among the four scenarios considered in this work and applying the corresponding model to minimize the error at parameters inference.

CRediT authorship contribution statement

Òscar Garibo-i-Orts: Data curation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Carlos Lizama:** Conceptualization, Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Ali Akgül:** Conceptualization, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **J. Alberto Conejero:** Conceptualization, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing, Formal analysis, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

JAC acknowledges funding from the Spanish Ministry of Science and Innovation (MICINN), grant PID2021-124618NB-C21 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe", by the "European Union".

CL is partially funded by Agencia Nacional para la Innovación y del Desarrollo (ANID), under FONDECYT grant number 1220036.

We also acknowledge funding for open access charge: CRUE-Universitat Politècnica de València.

References

[1] B. Kuttner, On differences of fractional order, Proc. Lond. Math. Soc. 3 (1) (1957) 453–466.
[2] J. Diaz, T. Osler, Differences of fractional order, Math. Comp. 28 (125) (1974) 185–202.
[3] H. Gray, N. Zhang, On a new definition of the fractional difference, Math. Comp. 50 (182) (1988) 513–529.
[4] K. Miller, B. Ross, Fractional difference calculus, in: Proceedings of the International Symposium on Univalent Functions, Fractional Calculus and their Applications, 1989, pp. 139–152.
[5] D. Baleanu, Z. Güvenç, J.T. Tenreiro-Machado, et al., New Trends in Nanotechnology and Fractional Calculus Applications, vol. 10, Springer, 2010.
[6] J.A. Conejero, J. Franceschi, E. Picó-Marco, Fractional vs. ordinary control systems: what does the fractional derivative provide? Mathematics 10 (15) (2022) 2719.
[7] M. Ortigueira, J.T. Tenreiro-Machado, Which derivative? Fractal Fract. 1 (1) (2017) 3.
[8] M. Tavazoei, Fractional order chaotic systems: history, achievements, applications, and future challenges, EPJ-Spec. Top. 229 (2020) 887–904.
[9] M. Khan, J.L. García-Guirao, Riemann Liouville fractional-like integral operators, convex-like real-valued mappings and their applications over fuzzy domain, Chaos Solitons Fractals 177 (2023) 114196.
[10] R. May, Simple mathematical models with very complicated dynamics, Nature 261 (5560) (1976) 459–467.
[11] G.-C. Wu, D. Baleanu, Discrete fractional logistic map and its chaos, Nonlinear Dynam. 75 (2014) 283–287.
[12] J.A. Conejero, C. Lizama, A. Mira-Iglesias, C. Rodero, Visibility graphs of fractional Wu–Baleanu time series, J. Difference Equ. Appl. 25 (9–10) (2019) 1321–1331.
[13] G. Muñoz-Gil, M.A. Garcia-March, C. Manzo, A. Celi, M. Lewenstein, Diffusion through a network of compartments separated by partially-transmitting boundaries, Front. Phys. 7 (2019) 31.
[14] G. Muñoz-Gil, M.A. Garcia-March, C. Manzo, J.D. Martín-Guerrero, M. Lewenstein, Single trajectory characterization via machine learning, New J. Phys. 22 (1) (2020) 013010.
[15] G. Muñoz-Gil, G. Volpe, M.A. Garcia-March, E. Aghion, A. Argun, et al., Objective comparison of methods to decode anomalous diffusion, Nature Commun. 12 (1) (2021) 6253.
[16] G. Muñoz-Gil, G. Volpe, M.A. García-March, R. Metzler, M. Lewenstein, C. Manzo, The Anomalous Diffusion challenge: objective comparison of methods to decode anomalous diffusion, in: Emerging Topics in Artificial Intelligence (ETAI) 2021, Vol. 11804, SPIE, 2021, 1180416.
[17] H. Seckler, J. Szwabiński, R. Metzler, Machine-learning solutions for the analysis of single-particle diffusion trajectories, J. Phys. Chem. Lett. 14 (35) (2023) 7910–7923.
[18] A. Argun, G. Volpe, S. Bo, Classification, inference and segmentation of anomalous diffusion with recurrent neural networks, J. Phys. A - Math. Theor. 54 (29) (2021) 294003.
[19] S. Bo, F. Schmidt, R. Eichhorn, G. Volpe, Measurement of anomalous diffusion using recurrent neural networks, Phys. Rev. E 100 (1) (2019) 010102.
[20] Ò. Garibo-i Orts, A. Baeza-Bosca, M.A. Garcia-March, J.A. Conejero, Efficient recurrent neural network methods for anomalously diffusing single particle short and noisy trajectories, J. Phys. A - Math. Theor. 54 (50) (2021) 504002.

[21] N. Firbas, Ò. Garibo-i Orts, M.A. Garcia-March, J.A. Conejero, Characterization of anomalous diffusion through convolutional transformers, J. Phys. A - Math. Theor. 56 (1) (2023) 014001.
[22] A. Gentili, G. Volpe, Characterization of anomalous diffusion classical statistics powered by deep learning (CONDOR), J. Phys. A - Math. Theor. 54 (31) (2021) 314003.
[23] A. Kumar, S. Das, S. Singh, et al., Quasi-projective synchronization of inertial complex-valued recurrent neural networks with mixed time-varying delay and mismatched parameters, Chaos Solitons Fractals 166 (2023) 112948.
[24] W. Du, Q. Miao, L. Tong, Y. Tang, Identification of fractional-order systems with unknown initial values and structure, Phys. Lett. A 381 (23) (2017) 1943–1949.
[25] T. Zhang, Z.-R. Lu, J.-K. Liu, G. Liu, Parameter estimation of fractional chaotic systems based on stepwise integration and response sensitivity analysis, Nonlinear Dynam. 111 (16) (2023) 15127–15144.
[26] G.-C. Wu, J.-L. Wei, T.-C. Xia, Multi-layer neural networks for data-driven learning of fractional difference equations' stability, periodicity and chaos, Phys. D: Nonlinear Phenom. 457 (2024) 133980.
[27] Z.-Q. Wu, G.-C. Wu, W. Zhu, Neural network method for parameter estimation of fractional discrete-time unified systems, Fractals 32 (01) (2024) 1–8.
[28] J.A. Conejero, Ò. Garibo-i Orts, C. Lizama, Inferring the fractional nature of Wu Baleanu trajectories, Nonlinear Dynam. 111 (2023) 12421–12431.
[29] J.A. Conejero, Ò. Garibo-i Orts, C. Lizama, Recovering discrete delayed fractional equations from trajectories, Math. Methods Appl. Sci. (2023) http://dx.doi.org/10.1002/mma.9228, In press.
[30] D. Baleanu, Y. Karaca, J.L. Vázquez, J. Macías-Díaz, Advanced fractional calculus, differential equations and neural networks: analysis, modeling and numerical computations, Phys. Scr. 98 (11) (2023) 110201.
[31] M. Joshi, S. Bhosale, V. Vyawahare, A survey of fractional calculus applications in artificial neural networks, Artif. Intell. Rev. (2023) 1–54.
[32] H. Niu, Y. Chen, B. West, Why do big data and machine learning entail the fractional dynamics? Entropy 23 (3) (2021) 297.
[33] B. Shiri, H. Kong, G.-C. Wu, C. Luo, Adaptive learning neural network method for solving time–fractional diffusion equations, Neural Comput. 34 (4) (2022) 971–990.
[34] J.-L. Wei, G.-C. Wu, B.-Q. Liu, J. Nieto, An optimal neural network design for fractional deep learning of logistic growth, Neur. Comput. Appl. 35 (15) (2023) 10837–10846.
[35] P. Meyer, R. Metzler, Stochastic processes in a confining harmonic potential in the presence of static and dynamic measurement noise, New J. Phys. 25 (6) (2023) 063003.
[36] H. Seckler, R. Metzler, Bayesian deep learning for error estimation in the analysis of anomalous diffusion, Nature Commun. 13 (1) (2022) 6717.
[37] CMS Collaboration, New developments for jet substructure reconstruction in CMS, detector performance summary: CMS-dps-17-027, 2017, https://cds.cern.ch/record/2275226.
[38] N. Geneva, N. Zabaras, Multi-fidelity generative deep learning turbulent flows, Found. Data Sci. 2 (4) (2020) 391–428.
[39] R. Maulik, R. Egele, B. Lusch, P. Balaprakash, Recurrent neural network architecture search for geophysical emulation, in: SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, 2020, pp. 1–14.
[40] D. Schmekel, F. Alcántara-Ávila, S. Hoyas, R. Vinuesa, Predicting coherent turbulent structures via deep learning, Front. Phys. 10 (2022) 309.
[41] N. Geneva, N. Zabaras, Transformers for modeling physical systems, Neur. Net. 146 (2022) 272–289.
[42] C. Andersson, A.H. Ribeiro, K. Tiels, N. Wahlström, T. Schön, Deep convolutional networks in system identification, in: 2019 IEEE 58th Conference on Decision and Control, CDC, IEEE, 2019, pp. 3670–3676.
[43] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-C. Woo, Convolutional lstm network: A machine learning approach for precipitation nowcasting, Adv. Neur. Inform. Proess. Syst. 28 (2015).
[44] M. Lozano, Ò. Garibo i Orts, E. Piñol, M. Rebollo, K. Polotskaya, M.A. Garcia-March, J.A. Conejero, F. Escolano, N. Oliver, Open data science to fight COVID-19: winning the 500k XPRIZE pandemic response challenge, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2021, pp. 384–399.
[45] J. Wang, T. Sun, B. Liu, Y. Cao, H. Zhu, CLVSA: A convolutional LSTM based variational sequence-to-sequence model with attention for predicting trends of financial markets, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI'19, AAAI Press, 2019, pp. 3705–3711.
[46] I. Shilon, M. Kraus, M. Büchele, K. Egberts, T. Fischer, T.L. Holch, T. Lohse, U. Schwanke, C. Steppa, S. Funk, Application of deep learning methods to analysis of imaging atmospheric Cherenkov telescopes data, Astropart. Phys. 105 (2019) 44–53.
[47] R. Abbasi, Y. Abe, T. Abu-Zayyad, M. Allen, Y. Arai, R. Arimura, E. Barcikowski, J. Belz, D. Bergman, S. Blake, et al., The energy spectrum of cosmic rays measured by the Telescope Array using 10 years of fluorescence detector data, Astropart. Phys. 151 (2023) 102864.
[48] Y. Wang, Y. Zhang, G.-G. Wang, Forecasting ENSO using convolutional LSTM network with improved attention mechanism and models recombined by genetic algorithm in CMIP5/6, Inform. Sci. 642 (2023) 119106.
[49] S. Kamal, C.S. Chandran, M. Supriya, Passive sonar automated target classifier for shallow waters using end-to-end learnable deep convolutional LSTMs, Eng. Sci. Technol. 24 (4) (2021) 860–871.
[50] D. Müller, U. Netzelmann, B. Valeske, Defect shape detection and defect reconstruction in active thermography by means of two-dimensional convolutional neural network as well as spatiotemporal convolutional LSTM network, Quant. Infrared Thermogr. J. 19 (2) (2022) 126–144.
[51] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[52] C. Manzo, M. Garcia-Parajo, A review of progress in single particle tracking: from methods to biophysical insights, Rep. Progr. Phys. 78 (12) (2015) 124601.