Anomalous Diffusion Characterization using Machine Learning Methods

PhD DISSERTATION

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Author
Òscar Garibo i Orts

València, February 2023

Advisors
Dr. José Alberto Conejero Casares
Dr. Miguel Ángel García March

A ma mare i mon pare,
per educar-me en llibertat.
A ma germana, Joel i León,
per fer gran la família.
A Guillem, t'estime.
A l'altra família, la que tries,
perquè us ho teniu guanyat.

Table of Contents

Abstract

I'd far rather be happy than right any day.

Douglas Adams, The Hitchhiker's Guide to the Galaxy.

During the last decades the use of machine learning and artificial intelligence has shown an exponential growth in many areas of science. The fact that computer's hardware has increased its performance while lowering the price and the availability of open source frameworks have enabled the access to artificial intelligence to a broad range of researchers, hence democratizing the access to artificial intelligence methods to the research community. It is our belief that multi-disciplinarity is the key to new achievements, with teams composed of researchers with different backgrounds and fields of specialization. With this aim, we focused this thesis in using machine learning, artificial intelligence, deep learing, all of them being understood as part of a whole concept we concrete in artificial intelligence, to try to shed light to some problems from the fields of mathematics and physics.

A deep learning architecture was developed and successfully benchmarked with the characterization of anomalous diffusion processes. Whereas traditional statistical methods had previously been used with this aim, deep learning methods, mainly based on recurrent neural networks have proved to outperform these clasical methods. Our architecture showed it can precisely infer the anomalous diffusion exponent and accurately classify trajectories among a given set of underlaying diffusion models.

While recurrent neural networks irrupted in the recent years, convolutional network based models had been extensively tested in the field of image processing for more than 15 years. There exist many models and architectures, pre-trained and set to be used by the community. No further investigation needs to be done since the architecture has proved its value for years and is very well documented in the literature. Our goal was being able to used this well-known and reliable models with anomalous diffusion trajectories. We only needed to be able to convert a time series into an image, which we successfully did by applying gramian angular fields to the trajectories, focusing on short ones. To our knowledge this is the first time these techniques were used in this field. We show how this approach outperforms any other proposal in the underlaying diffusion model classification for short trajectories.

Besides physics it is maths. We used our recurrent neural networks architecture to infer the parameters that define the Wu Baleanu trajectories. We show that our proposal can precisely infer both the μ and ν parameters with a reasonable confidence. Being the first time, to the best of our knowledge, that such techniques were applied to this scenario. We extend this work to the discrete delayed fractional equations, obtaining similar results in terms of precision. Additionally, we showed that the same architecture can be used to discriminate delayed from non-delayed trajectories with a high confidence.

Finally, we also searched fractional discrete models. We have considered Lubich's quadrature time-stepping schemes instead of the classical Euler scheme of order 1. As the first study with this new paradigm, we compare the bifurcation diagrams for the logistic and sine maps obtained from Euler discretizations of orders 1, 2, and ½.

Resumen

Durante las últimas décadas el uso del aprendizaje automático (*machine learning*) y de la inteligencia artificial ha mostrado un crecimiento exponencial en muchas áreas de la ciencia. El hecho de que los ordenadores hayan aumentado sus prestaciones a la vez que han reducido su precio, junto con la disponibilidad de entornos de desarrollo de código abierto han permitido el acceso a la inteligencia artificial a un gran rango de investigadores, democratizando de esta forma el acceso a métodos de inteligencia artificial a la comunidad investigadora. Es nuestra creencia que la multidisciplinaridad es clave para nuevos logros, con equipos compuestos de investigadores con diferentes preparaciones y de diferentes campos de especialización. Con este ánimo, hemos orientado esta tesis en el uso de *machine learning*, inteligencia artificial, aprendizaje profundo o *deep learning*, entendiendo todas las anteriores como parte de un concepto global que concretamos en el término inteligencia artificial, a intentar arrojar luz a algunos problemas de los campos de las matemáticas y la física.

Desarrollamos una arquitectura *deep learning* y la medimos con éxito en la caracterización de procesos de difusión anómala. Mientras que previamente se habían utilizado métodos estadísticos clásicos con este objetivo, los métodos de *deep learning* han demostrado mejorar las prestaciones de dichos métodos clásicos. Nuestra architectura demostró que puede inferir con precisión el exponente de difusión anómala y clasificar trayectorias entre un conjunto dado de modelos subyacentes de difusión .

Mientras que las redes neuronales recurrentes irrumpieron recientemente, los modelos basados en redes convolucionales han sido ámpliamente testados en el campo del procesamiento de imagen durante más de 15 años. Existen muchos modelos y arquitecturas, pre-entrenados y listos para ser usados por la comunidad. No es necesario realizar investigación ya que dichos modelos han probado su valía durante años y están bien documentados en la literatura. Nuestro objetivo era ser capaces de usar esos modelos bien conocidos y fiables, con trayectorias de difusión anómala. Solo necesitábamos convertir una serie temporal en una imagen, cosa que hicimos aplicando *gramian angular fields* a las trayectorias, poniendo el foco en las trayectorias cortas. Hasta donde sabemos, ésta es la primera vez que dichas técnicas son usadas en este campo. Mostramos cómo esta aproximación mejora las prestaciones de cualquier otra propuesta en la clasificación del modelo subyacente de difusión anómala para trayectorias cortas.

Más allá de la física están las matemáticas. Utilizamos nuestra arquitectura basada en redes recurrentes neuronales para inferir los parámetros que definen las trayectorias de Wu Baleanu. Mostramos que nuestra propuesta puede inferir con razonable precisión los parámetros μ y ν. Siendo la primera vez, de nuevo hasta donde llega nuestro conocimiento, que tales técnicas se aplican en este escenario. Extendemos este trabajo a las ecuaciones fraccionales discretas con retardo, obteniendo resultados similares en términos de precisión. Adicionalmente, mostramos que la misma arquitectura se puede usar para discriminar entre trayectorias con y sin retardo con gran confianza.

Finalmente, también investigamos modelos fraccionales discretos. Hemos analizado esquemas de paso temporal con la cuadratura de Lubich en lugar del clásico esquema de orden 1 de Euler. En el primer estudio de este nuevo paradigma hemos comparado los diagramas de bifurcación de los mapas logístico y del seno, obtenidos de la discretización de Euler de orden 1, 2 y ½.

Resum

Durant les darreres dècades l'ús de l'aprenentatge automàtic (*machine learning*) i de la intel.ligència artificial ha mostrat un creixement exponencial en moltes àrees de la ciència. El fet que els ordinadors hagen augmentat les seues prestacions a la vegada que han reduït el seu preu, junt amb la disponibilitat d'entorns de desenvolupament de codi obert han permè l'accés a la intel.ligència artificial a un gran rang d'investigadors, democratitzant així l'accés a mètodes d'intel.ligència artificial a la comunitat investigadora. És la nostra creença que la multidisciplinaritat és clau per a nous èxits, amb equips compostos d'investigadors amb diferents preparacions i diferents camps d'especialització. Amb aquest ànim, hem orientat aquesta tesi en l'ús d'intel.ligència artificial, *machine learning*, aprenentatge profund o *deep learning*, entenent totes les anteriors com a part d'un concepte global que concretem en el terme intel.ligència artificial, a intentar donar llum a alguns problemes dels camps de les matemàtiques i la física.

Desenvolupem una arquitectura *deep learning* i la mesurem amb èxit en la caracterització de processos de difusió anòmala. Mentre que prèviament s'havien utilitzat mètodes estadístics clàssics amb aquest objectiu, els mètodes de *deep learning* han demostrat millorar les prestacions d'aquests mètodes clàssics. La nostra architectura va demostrar que pot inferir amb precisió l'exponent de difusió anòmala i classificar trajectòries entre un conjunt donat de models subjacents de difusió.

Mentre que les xarxes neuronals recurrents van irrompre recentment, els models basats en xarxes convolucionals han estat àmpliament testats al camp del processament d'imatge durant més de 15 anys. Hi ha molts models i arquitectures, pre-entrenats i llestos per ser usats per la comunitat. No cal fer recerca ja que aquests models han provat la seva vàlua durant anys i estan ben documentats a la literatura. El nostre objectiu era ser capaços de fer servir aquests models ben coneguts i fiables, amb trajectòries de difusió anòmala. Només necessitàvem convertir una sèrie temporal en una imatge, cosa que vam fer aplicant *gramian angular fields* a les trajectòries, posant el focus a les trajectòries curtes. Fins on sabem, aquesta és la primera vegada que aquestes tècniques són usades en aquest camp. Mostrem com aquesta aproximació millora les prestacions de qualsevol altra proposta a la classificació del model subjacent de difusió anòmala per a trajectòries curtes.

Més enllà de la física hi ha les matemàtiques. Utilitzem la nostra arquitectura basada en xarxes recurrents neuronals per inferir els paràmetres que defineixen les trajectòries de Wu Baleanu. Mostrem que la nostra proposta pot inferir amb raonable precisió els paràmetres μ i ν. Sent la primera vegada, novament fins on arriba el nostre coneixement, que aquestes tècniques s'apliquen en aquest escenari. Estenem aquest treball a les equacions fraccionals discretes amb retard, obtenint resultats similars en termes de precisió. Adicionalment, mostrem que la mateixa arquitectura es pot fer servir per discriminar entre trajectòries amb i sense retard amb gran confiança.

Finalment, també investiguem models fraccionals discrets. Hem analitzat esquemes de pas temporal amb la quadratura de Lubich en lloc del clàssic esquema d'ordre 1 d'Euler. Al primer estudi d'aquest nou paradigma hem comparat els diagrames de bifurcació dels mapes logístic i del sinus, obtinguts de la discretització d'Euler d'ordre 1, 2 i ½.

1 Introduction

> *When Fortuna spins you downward, go out to*
> *a movie and get more out of life.*
>
> John Kennedy Toole, A Confederacy Of Dunces.

1.1 Motivation

It was in 1826 that the Scottish botanist Robert Brown was studying a new plant, *Clarkia pulchella* when he observed that some particles that came from the plant's pollen grains followed an erratic and continous motion when suspended in water. In order to discard biological sinergies between the water and the particles that came from pollen grains, Brown conducted a similar experiemnt but using inorganic dust particles. In this new experiment, he observed exactly the same behaviour. It was not until 80 years later when Albert Einstein and Marian Smoluchowski gave the theoretical framework to explain such behaviour. In plain words, what Einstein and Smoluchowski proposed was an extension of the thermodynamics of molecular gases and liquids, which present a random motion, to solids. Even that they did not have the experimental tools to prove it, they theorized that atomic fluctuations were able to affect other bodies. That is, atoms and mollecules colliding with other bodies affect the motion of these other bodies, even if they are big, with this collisions being stochastic, as it was the affection to the bodies.

Brownian Motion presents an anomalous diffusion exponent $\alpha = 1$, and the mean squared displacement (MSD), or the variance of the displacements' length, grows linearly with time. But besides Brownian Motion, many other underlaying models have been described. Models such as annealed transient time motion, continous time random walk, and other. They differ from Brownian Motion in the possible values of α, which determines if the models are subdiffusive, superdiffusive, balistic or Brownian. The different models usually have different behaviours even if they share the same value for α. This behaviour can be determined, among others, by the underlaying sustrate where the motion takes place. So, two main problem arise when working with anomalous diffusing trajectories: inferring the anomalous exponent α, and identifying the underlaying model that characterizes the trajectory.

During the last decade of last century, a machine learning revolution took place. Despite most of the mathematical models and procedures had been available for years, by this time the computational resources achieved the required performance to implement those methods. An increasing number of neural networks, which pretend to emulate the behaviour of the neurons in human brains, were developed. Convolutional neural networks (CNN) were the first to show up, mainly with application to the image processing field. Since CNNs were not able to properly process temporal series data, recurrent neural networks (RNN) were developed. The first RNN underwent problems when processing long temporal series, then Long Short Term Memory (LSTM) neural networks emerged to fix the long term memory issues. And the process is still going on, with the appearance of Transformers, Generative Adversarial Networks (GAN) and many other.

The main goal of this thesis is to explore the use of these machine learning methods to shed light to physical processes like anomalous diffusion. The possibility of implementing a method to learn how to characterize a given anomalous diffusing trajectory, that is inferring its anomalous exponent α and determining the underlaying model is of high interest. Mainly because no previous knowledge about the physics inherent to the anomalous diffusion is required. After tackling this problem the natural path is exploring other fields in which the provided solution could be applied, for instance epidemiological predictions or mathematical problems. Which lead to propose the following research questions:

RQ1 - Can anomalous diffusion behaviour be modeled without having any prior knowledge? Is it possible to avoid using statistical methods which require in depth pyshics knowledge?

RQ2 - May machine learning methods use raw trajectories to infer the anomalous exponeent α and determine the underlaying diffusion model?

RQ3 - Could we extend the use of these machine learning methods to other problems in mathematics and physics fields?

The research work developed in this thesis aims to answer these questions. Machine learning based models were developed to characterize anomalous diffusing particles in two senses. On the one hand, the models predicted the anomalous exponent α and, in the other hand, the models classified the underlaying model generating the anomalous diffusing trajectory among a set of possible candidates or classes. Besides that, the same architecture (with some adjustements in some cases) was used in completely different scenarios such us epidemiological temporal series data, or mathematical problems. To this end, the following objectives were defined:

O1 - Review state-of-the-art in regards of machine learning methods applied to anomalous diffusion characterization.

O2 - Review state-of-the-art of machine learning methods capable to deal with time series.

O3 - Participate in the AnDi Challenge to benchmark our proposed architecture with the worlwide top solutions in the field.

O4 - Evaluate the proposed architecture with completely different temporal data using epidemiological data related to the COVID-19 pandemic.

O5 - Explore the use of traditional image processing methods applied to time series data by applying Gramian Angular Fields transformation.

O6 - Extend the application of the developed methods to other mathematical scenarios, specifically the fractional logistic.

The aforementioned objectives enclose the main goal of this thesis: *the study of how machine learning methods can help in the characterization of anomalous diffusion processes.* Additionally, the objectives encompass other interesting applications of the product of the research, that is, the use of machine learning to tackle time series data. The following scientific contributions support the achievement of the proposed objectives:

1.2 Thesis contributions

This section presents the main contributions of this thesis. First, a succint presentation of the most relevant aspects of each contribution is done. Next, the scientific publications in high impact factor journals are presented.

1.2.1 Main contributions

C1 - **Characterization of anomalous diffusion using machine learing methods**
In this work we present the machine learning based architecture developed to precisely infer the anomalous diffusion exponent α and accurately assign the underlaying model generating an anomalous diffusing trajectory. The architecture was benchmarked in the first international competition (AnDi Challenge, for Anomalous Diffusion Challenge) where the models achieved honors by the performance they showed.

C2 - **Gramian Angular Fields for leveraging pre-trained computer vision models with anomalous diffusion trajectories**.
Gramian Angular Fields allows the representation of a time series as an image without loosing temporal information, which allows the use of well-known image processing CNN models. To the best of our knowledge, in this work we present the first application of Gramian Angular Fields to characterize anomalous diffusion taking advantage of CNN based architectures which have an outstanding performance with image classification and processing.

C3 - **Benchmarking with epidemiological data: XPrize Pandemic Response Challenge**
The COVID-19 pandemic arised as this work was in progress. A group of researchers from Comunitat Valenciana enrolled the XPrize Pandemic Response Challenge as the ValenciaIA4Covid team, which won the challenge. The architecture proposed in this work (with slightly modifications) was used to predict daily COVID-19 cases in 236 countries or regions.

C4 - **Inferring fractional logistic parameters**
The architecture proposed in this work has also been used to shed light to the fractional nature of the Wu-Baleanu trajectories. It has been possible to infer the μ and ν paremeters, which define the fractional behaviour from unseen trajectories with a reasonable performance. To the best of our knowledge, this has been the first time a machine learning approach to this problem has been done. The same architecture was able to accurately classify fractional trajectories among Wu-Baleanu class, and fractional delayed class, showing its versatility.

1.2.2 Scientific Publications

The scientific contributions of this thesis have been published in four scientific top-ranked journals and two conference proceedings in the fields of Multidisciplinary Sciences, Clinical Neurology, Computer Science, Biomedical Engineering, Medical Informatics, and Applied Mathematics. All the following journal papers are either published, accepted or under review in journals included in the Journal Citation Reports (JCR). The versions presented in this dissertation are adaptations for the thesis due to university regulations. Each of them is covered as a chapter having the same structure and bibliography as the original published version:

P1 - **Òscar Garibo-i-Orts**, Alba Baeza-Bosca, Miguel Ángel García-March, José Alberto Conejero. "Efficient recurrent neural network methods for anomalously diffusing single particle short and noisy trajectories". *Journal of Physics A-Mathematical and Theoretical* (2021) 54(50).

IF: 2.331 (JCR 2021): 14/56 (Q1) Physics, Mathematical, 45/86 (Q2) Physics, Multidisciplinary.

P2 - **Òscar Garibo-i-Orts**, Nicolás Firbas, Laura Sebastiá, José Alberto Conejero. "Gramian Angular Fields for leveraging pre-trained computer vision models with anomalous diffusion trajectories". *Physical Review E. Accepted for publication*

IF: 2.707 (JCR 2021): 10/56 (Q1) Physics, Mathematical, 16/34 (Q2) Physics, Fluids & Plasmas.

P3 - José Alberto Conejero, **Òscar Garibo-i-Orts**, Carlos Lizama. "Inferring the fractional nature of Wu Baleanu trajectories". *Nonlinear Dynamics.* (November 2022) *Accepted for publication*

IF: 5.741 (JCR 2021): 14/137 (Q1) Engineering, Mechanical, 13/138 (Q1) Mechanics.

P4 - José Alberto Conejero, **Òscar Garibo-i-Orts**, Carlos Lizama. "Recovering discrete delayed fractional equations from trajectories". *Mathematical Methods in the Applied Sciences. Under revision*

IF: 3.007 (JCR 2021): 29/267 (Q1) Mathematics/Applied.

P5 - José Alberto Conejero, **Òscar Garibo-i-Orts**, Carlos Lizama. "On a new paragigm in the Logistic and similar maps: time-stepping schemes". *Preprint*

1.3 Outline

This thesis is structured in nine chapters describing the research work carried out during the stage of development of this dissertation. Chapter 1 introduces the motivation of this thesis, the research objectives, and the main contributions. Chapter 2 presents the machine learning architecture used in the compendium of articles included in this thesis. Chapter 3 describes the participation in the Anomalous Diffusion Challenge, the real embryo of this Thesis. Chapter 4 introduces the contribution made within the ValenciaIA4Covid team, winner of the XPRIZE Pandemic Response Challenge during the COVID-19 pandemics and its relevance. Chapter 5 describes how our architecture, ConvLSTM can capture the particularities of anomalous diffusing trajectories for anomalous diffusion exponent inference and trajectory classification. Chapter 6 proposes Gramian Angular Field based image representation for anolomalous diffusing trajectories and benchmarks the usage of state-of-the-art image models with short trajectories. Chapter 7 extends the application of our model to infer the fractional nature of Wu-Baleanu trajectories. Chapter 8 introduces the usage of ConvLSTM to retrieve the nature of fractional equation from trajectories, and its ability to discriminate delayed from non-delayed trajectories. Finally, chapter 9 presents the concluding remarks and proposes guidance to continue with the research developed in this thesis.

2 ConvLSTM

La única lucha que se pierde es la que se abandona.

Ernesto Che Guevara.

This thesis shares the objectives of the *design science* research paradigm, developing knowledge that other professionals (let them be physicists, mathematicians, epidemiologists or even decission makers) can use to design solutions for problems in their research fields. This is an engineer-like approach which uses existing tools to build new approaches to some known problems.

The main tool used in this thesis is machine learning, in particular deep learning. In this chapter machine learning is introduced, including required definitions to put them in the context of this thesis. The architecture used is introduced and explained in detail in the following sections. First, a brief introduction to machine learning is presented. Secondly, the deep learning methods used in this thesis are introduced. Finally, the proposed architecture that has been used in this work is presented.

2.1 Machine learning

As a starting definition, one can say that Machine learning is a discipline that builds algorithms which are capable to learn from data. This relies in the assumption that procedures which worked well in the past will, hopefully, also work well in the future. A dumb example can be: as sun has been rising in the morning for the last 10,000 years, it is likely true that it will rise tomorrow morning and all mornings in the future. That is, given some patterns that have been true in the past, one can expect the same patterns be true in the future. And that is, roughly speaking, what machine learning methods rely on, finding patterns in some amount of data, and apply these patterns to unseen data. These methodology has widely been used by humans along history. Humans have been able to find algorithms to extract these patterns, generally using (rules based) algorithms. But the human eye (or grain) fails to find these patterns when data is very numerous, or complex, involving millions or billions of records, or when dealing with high dimensional data. As a first approximation, humans were able to write algorithms which were executed by machines (Pascal calculator, earlier computers), thus being able to process bigger amounts of data than a human could process, and not of less importance, processing this information in a more rapid way.

Going to an extensive description of machine learning as a discipline is not a purpose of this thesis, nor getting the insight of the complete range of methods and approaches. But a brief introduction is given of the main approaches, while deeper insights are offered in regards of the methods used in this work.

Before diving into the machine learning methods, some definitions are needed and terminology explained for the sake of understandability. The purpose of machine learning is building a model which is capable of learning from available data and thus evaluate unseen data. To do so, available data that comes from an unknown probability distribution (hopefully representative of the space of occurrences) has to be divided into three main data sets: 1) *training*, 2) *validation* and 3) *test* data sets. 1) *Training* data set, the machine learning method builds a model from this data set, and this model should generalize to the space of occurrences, so it can produce accurate predictions for unseen data. 2) *Validation* data set, which is used during the learning process to validate the model as it learns. And 3) *test* data set, is used to evaluate the model's performance once the learning process is finished. The test data set is of ultimate importance since this data set should never be seen by the model during the training process and be representative (ideally) of the space of recurrences the model has to generalize to. In fact, this is the best possible way to evaluate what the behaviour of the model will be with all the existing, and to exist, unseen data.

Depending on the desired output of the algorithm or machine learning method, three main categories can be described:

- **Classification:** the data is split into two or more classes or groups. Once the machine learning model is trained and validated, it can assign unseen data to one of the classes it has seen during the training process. No need to say that if a new class arises and the model has not seen this class, none of the data belonging to the new class (or classes) will be correctly assigned. As new classes show, the model will need of further training to learn the patterns defining this new class. A classification model is evaluated using metrics like accuracy ($\frac{\text{correct classifications}}{\text{all classifications}}$) or the f1-score (see definition in Chapter 3, section 3.2 The Challenge: Evaluation).
- **Regression:** the desired output for the model belongs to the continuous space of $\mathbb{R}$. So, the model will learn from training and validation data sets a function to fit to the values given during training. The model, as exposed for classification, is evaluated against the test data set, usually using metrics like mean averaged error (MAE, the average of the errors made for each sample), mean squared error (MSE, the mean of the squared errors made for each sample) or root mean square error (RMSE, or the square root of the MSE).
- **Clustering:** in this case the data is not split into different data sets since the goal of clustering methods is to divide the data into a known or unknown number of groups, so-called clusters. Data is represented in some n-dimensional space (being n the number of features defining each sample) and the model has to group the data according to a defined distance metric, usually the euclidean distance.

According to how the machine learing algorithms or methods learn, they can be divided in two main categories:

- **Supervised learning:** supervised algorithms need the data (training, validation and test) to be labelled, either with the class they belong to (in the case of classification) or the value they correspond to (in the case of regression). Technically it is possible to perform clustering with labelled data, but the information clustering could provide is already present in the data labels and therefore we do not consider clustering in the supervised learning.
- **Unsupervised learning:** unsupervised algorithms, such us clustering, learn patterns from the training data (the complete available data set) without having any reference in the form of label.

All the models used in this thesis were built using supervised training, both for classification and regression. During the development of this thesis we had to work with different types of time series. Some times in the form of successive points of a moving particle or object, thats is, a trajectory. Some others in the form of time dependent data along a time period, like daily Covid-19 cases. No matter what the information is representing, this information is time dependent, where the past has to be considered in order to correctly predict the next steps (which represents a position in some vectorial space, a number of Covid-19 cases in some country, a stock pricing, etc).

Recurrent neural networks (RNN) are one type of deep learning, so-called for the deep architectures that can be implemented by stacking layers over layers. RNN were created to deal with this kind of information, where not just the last available piece of information needs to be taken into account, but as much as possible back in the past. In the next sections we first introduce the

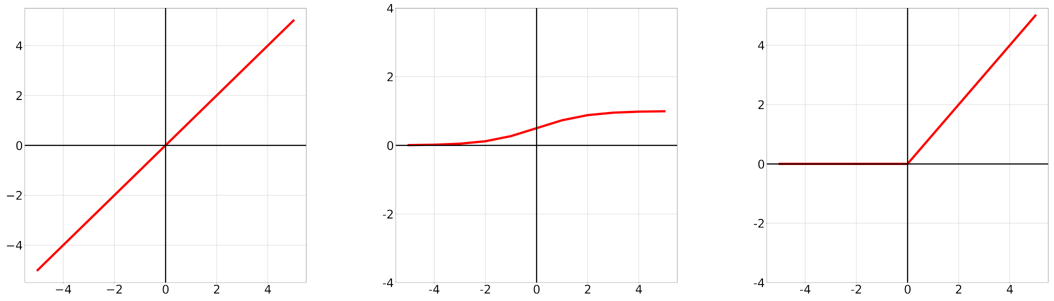


Fig. 2.1: Activation functions. Linear (left). Sigmoid (center). Rectified Linear Unit (ReLU) (right).

deep learning methods that form the architecture proposed in this thesis, and afterwards we give more details about the architecture we have used to address physics, mathematics and other kind of problems. We go through the architecture, justifying our decission making and evaluating the different alternatives.

2.2 Deep learning

Deep learning is a part of the machine learning methods based on artificial neural networks, which were designed to resemble how human (or other animals) brain works. As mentioned before, the main characteristic of deep learning is that multiple types of layers can be stacked sequentially (can be seen as one ontop the other), so the output of one layer feeds the input of the next layer. By mixing different kinds of layers, it is possible to achieve multiple different models with expertise in different tasks, and offering different levels of abstraction.

In this thesis 4 different types of layers have been used: densely connected layers, dropout layers, convolutional layers and long short term memory layers, which we introduce as follows:

2.2.1 Dense layer

A Dense is deeply connected to its previous layer, that is, every unit (or neuron) in a Dense layer is connected to every neuron in the preceding layer. Every neuron in the Dense layer has its own weight vector (or matrix) and each one performs a dot product between the input from the previous layer and its own weight vector, producing a result which if forwaded to the next layer. If two Dense layers are concatenated, each neuron from the first layer is connected to each neuron in the second layer, making them fully connected, and because of that receiving the name of Dense layer, since the density of connections is very high.

A Dense layer applies an activation function to its output, which determines the range of the output. Depending on the task to be performed different activation functions are required. In our architecture we use three different activation functions:

- Linear activation function: $f(x) = x$. Used at the final Dense layer for regression tasks.
- Sigmoid activation function: $f(x) = \frac{1}{1+e^{-x}}$. Used at the final Dense layer for classification tasks.
- Relu activation function: $f(x) = \max(0, \mathrm{x})$. Used in intermediate Dense layer because it is easy and cheap to compute.

The behaviour of these three activation functions is shown in Figure 2.1.

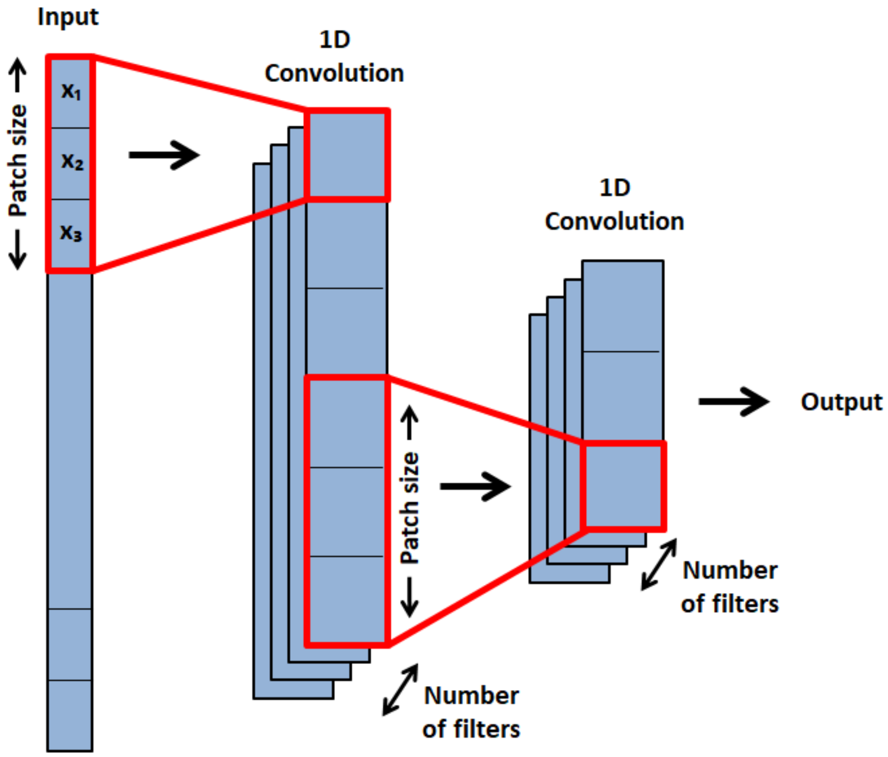


Fig. 2.2: Basic representation of a CNN.

2.2.2 Dropout layer

When using deep learning, a model is prone to fit the training data during the training process. That is, the model can specialize in the training data, losing its capabilities of generalizing with unseen data. In order to avoid such scenario, Dropout layers can be used. A Dropout layer randomly (with a given probability) removes connections between layers during training process. In other words, the dropout layer hides a defined portion of the information to the next layer in the deep learning model, preventing the model to overfit to the training data and easying the generalization of the model.

2.2.3 Convolutional layer

Convolutional neural networks (CNN) preserve the spatial structure of data by connecting a patch (or section) of data to a single neuron. Then, every neuron learns the properties from this single patch, whose size is defined by the kernel size. This way, spatially close portions of data are likely to be related and correlated to each other since only a small region of the input data influences the output of each neuron [1, 2]. The patch is slid across the input sequence becoming an sliding window, and each time we slide it, we have a new output neuron in the following layer. This lets us consider the spatial structure inherent to the input sequence [3, 4]. The number of patches per layer is defined as the number of filters, being each filter a patch with a different initialization for its weight's vector. Through these layers, we are able to learn trajectory features by weighting the connections between the patches and the neurons so that particular features can be extracted by each patch. In Figure 2.2 we depict the behaviour of a CNN and the patch sliding over it.

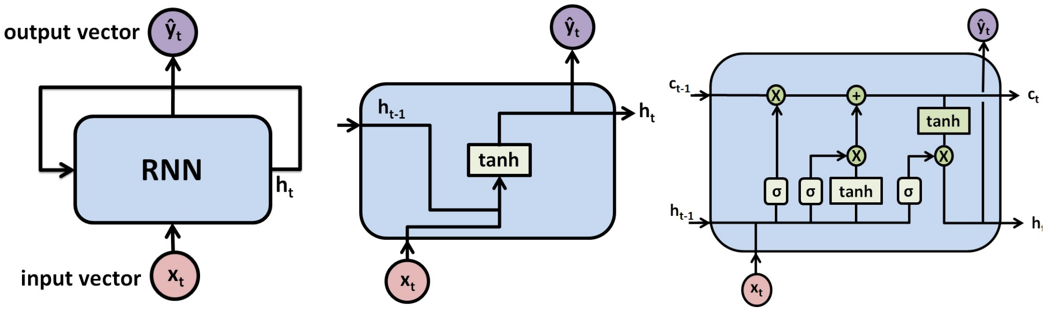


Fig. 2.3: Basic representation of a general RNN (left). An RNN with an inner tanh activation function (center). A scheme of an LSTM layer (right).

2.2.4 Long Short Term Memory Layers

Sequential information can be decomposed in single-time steps, such as words or characters in language, notes in music, codons in DNA sequences, etc. So, if one considers sequential data it is very likely that the output at a later time step will depend on the inputs at prior time steps. In practice, we need to relate the information from a particular time step also with prior time steps and pass this information to future times.

Recurrent neural networks (RNN) address this problem by adding an internal memory or cell state, denoted by h, which is passed from the time t to the time $t+1$, that is from h_t to h_{t+1}. This recurrent relation is capturing some notion of memory of what the sequence looks like. Therefore, the RNN output is not only a function of the input at a particular time step but also a function of the past memory of the cell state. In other words, the output $\overline{y_t} = f(x_t, h_{t-1})$, depends on the current input x_t and the previous outputs to the RNN h_{t-1}, as it can be seen Figure 2.3.

An RNN adapts the internal hidden state (or memory state) h_t through the result of multiplying two weight matrices W_{hh} and W_{xh} to the previous cell state h_{t-1} and the current input x_t, respectively. The weight matrix W_{hh} is modified at each time step to let the cell learn how to fit the desired output, and W_{xh} is the weight matrix that modules the contribution of the input at each time step to the learning process. The result is passed to an activation function *tanh* that modifies the current state at each time step, i.e. $h_t = \tanh\left(W_{hh}^T h_{t-1} + W_{xh}^T x_t\right)$.

The problem with RNNs arises when dealing with long sequences, since composing multiple tanh functions entails that the hidden state tends to extinguish by reaching values very close or equal to zero. In practice, this means that only recent cell states will modify the current cell state or, in other words, that RNNs have short-term memory.

Long short-term memory (LSTM) [5, 6] amend the aforementioned short-term memory problem implicit to RNN by including gated cells that allow them to maintain long-term dependencies in the data and to track information across multiple time steps. This improves the sequential data modeling. LSTM structure is shown in Figure 2.3 where σ and tanh stand for the sigmoid and the hyperbolic tangent activation functions The circles in red represent element-wise matrix multiplication and additions. An LSTM incorporates a new cell state channel c which can be seen as a transportation band where the information is selectively updated by the new gates and is independent of the previously defined hidden state h and, therefore, independent of what is outputted in the form of hidden state or current time step out.

One LSTM cell's composition can be seen in Figure 2.3 (right), and the gates are used to control the flow of information as follows:

- The first sigmoid gate decides what information is kept or rid of. Since the sigmoid output ranges from 0 to 1, this can be seen as a switch that modulates how much information from the previous state has to be kept.

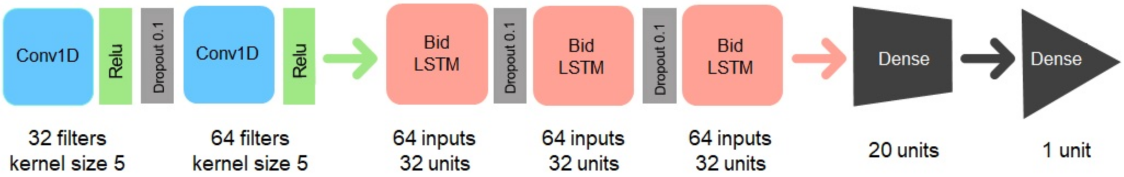


Fig. 2.4: Architecture of the model.

- The second gate, consisting of a sigmoid and a *tanh* functions store relevant information to the newly added cell state channel (c).
- Then, the outputs of the two previous gates are used to update the cell state (c) selectively.
- And the last sigmoid and tanh functions produce two different outputs; the new cell state (c), which is forwarded to the next LSTM cell, and the current time step output, which is a filtered version of the cell hidden state (h).

Further details about LSTM functioning and implementation can be found in [7, 8].

2.3 Architecture of the method

We propose the architecture shown in Figure 2.4 to learn from time series, whatever their characteristics are. The definition of the architecture has been extracted from our paper [9], which we include in this Thesis.

1. First, we have two convolutional layers that permit the extraction of spatial features from the trajectories. The first convolutional layer is set with 32 filters and a sliding window (kernel) of size 5, which slides through each trajectory extracting spatial features from them. The second convolutional layer has 64 filters to extract higher-level features.
2. Second, the output of the convolutional layers feeds three stacked bidirectional LSTMs layers that permit learning the sequential information. After each of these layers, we include a dropout layer of the 10% neurons to avoid over-fitting. We tested several dropout levels, from 5% to 20%, being 10% the one with the best performance.
3. Finally, we use two fully connected dense layers: the first one with 20 neurons and the second one with 1 or 2 neurons. This last choice depends if we want to predict a single parameter or both of them at the same time.

3 AnDi Challenge

Perquè hi haurà un dia que no podrem més
i llavors ho podrem tot.

Vicent Andrés Estellés.

The content of this section is partially extracted from the paper Muñoz-Gil et al., Nature Commun. 12, 10.1038 (2021) [10], of which the author of this thesis is co-author.

Back in 1827, the botanist Robert Brown observed on the microscope that small particles contained in the pollen of plants irregularly and fidgety moved while being suspended in water. This phenomenon drew his attention to the point of repeating the experiment with dust particles, showing the same result. He selected dust particles since he wanted to refute that the pollen particles' movement was not related to the pollen particles being alive. While several studies were contucted in this area, the physics to describe this behaviour still took almost 80 years to show. During the first decade of the 20th century Albert Einstein, Marian Smoluchowski and William Sutherland provided with the theoretical foundation for the processes Robert Brown's observations.

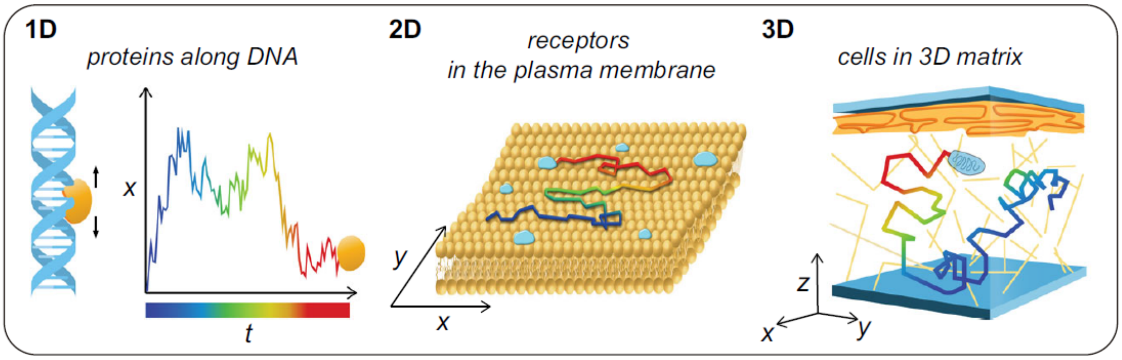


Fig. 3.1: Trajectories in spaces with different dimensionality. 1 dimension (1D), proteins sliding along DNA; 2 dimensions (2D), receptors diffusing in the plasma membrane; 3 dimensions (3D), cells migrating in a three dimensional matrix. Extracted from [10] with Creative Commons License `https://creativecommons.org/licenses/by/4.0/`

A random walk is characterized by an arbitrary change of some observable item over time, let it be a particle's position, some measurable temperature or even stock pricing, among many others, see Figure 3.1. In the same way, random walks happen at different scales (see Figure 3.2), from atomic scale to complex organism, in the diffusion of cellular components, even in bacteria and cells movement. They all share that their mean squared displacement (MSD) is not constant, but grows linearly with time (MSD α t), and it is known as normal diffusion. It has also been observed other kinds of diffusion that do not follow this distribution, but a power-law dependence (MSD αt^{α}) and we refer to this diffusion as anomalous diffusion, as opposed to normal diffusion. There are five possible scenarios following this anomalous diffusion which depend on the possible values of the diffusion exponent α:

- non mobility, when $\alpha = 0$,
- subdiffusion, when $0 < \alpha < 1$,
- normal diffusion or Brownian motion, when $\alpha = 1$,
- superdiffusion, when $1 < \alpha < 2$,
- balistic motion, when $\alpha \geq 2$.

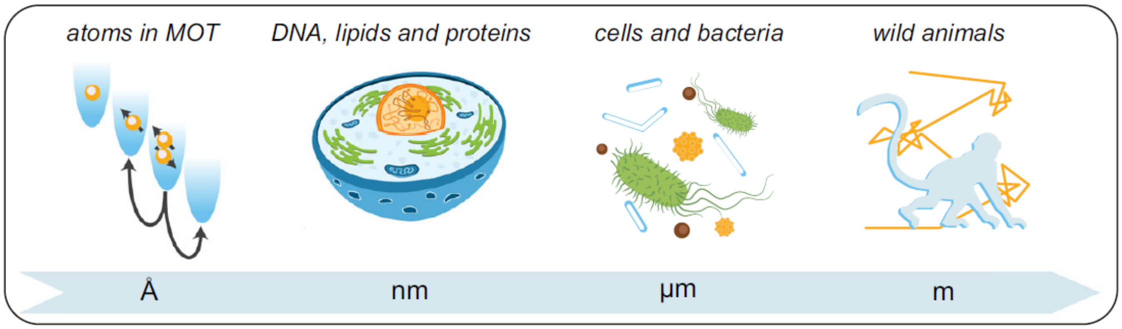


Fig. 3.2: Examples of random walks in (a) magneto-optical traps; (b) diffusion of cellular components (DNA, proteins, lipids, organelles); (c) the motion of bacteria and cells; and (d) animals. Extracted from [10] with Creative Commons License https://creativecommons.org/licenses/by/4.0/

When an experiment is performed to understand diffusion, one typically disposes of some trajectories from an observable item and there is the need to obtain information from these trajectories about the item itself and the medium where the trajectoiries take place, what in practice means inferring the anomalous diffusion exponent α, determining the underlaying diffusion model generating the trajectory, and determining if these properties show any change over space and time.

In order to tackle the inference of the anomalous diffusion exponeent α, we try to estimate the anomalous diffusion exponent α by fitting the MSD to a power law in the form of MSD αt^{α}. If there are multiple trajectories from multiple items, MSD is defined as the average of the items or ensemble average MSD (EA-MSD). If there are few but long trajectories, MSD can be calculated as a time average MSD over the full trajectory, or time-averaged MSD (TA-MSD), that is, cutting the long trajectory in a number of consecutive trajectories, computing the MSD and averaging them. If both (EA-MSD and TA-MSD) result in the same value, we say that the trajectories are ergodic, naming them non-ergodic if both values result different.

Determining the underlaying diffusion model is related to the driving physical mechanism. This is of great importance, since different models result in different trajectories even if they share the same exponent α.

And finally, the properties of trajectories can change over space and time because of changes in the environment, such as patches with different viscosity on a cellular membrane, or because of changes in the properties of the item the trajectory belongs to, for example having different activation states of a molecular motor). In order to get deeper insights in these cases it is interesting splitting the trajectories into smaller parts with different behaviours or characteristics, that is, segmenting the trajectory in subtrajectories and inferring the α and their underlaying models.

With all this in mind, scientists from several institutions around Europe joined efforts to launch the AnDi (AnDi stands for Anomalous Diffusion) Challenge in 2020 http://www.andi-challenge.org/, see Figure 3.3. These scientists came from reputed institutions as Institut de Ciències

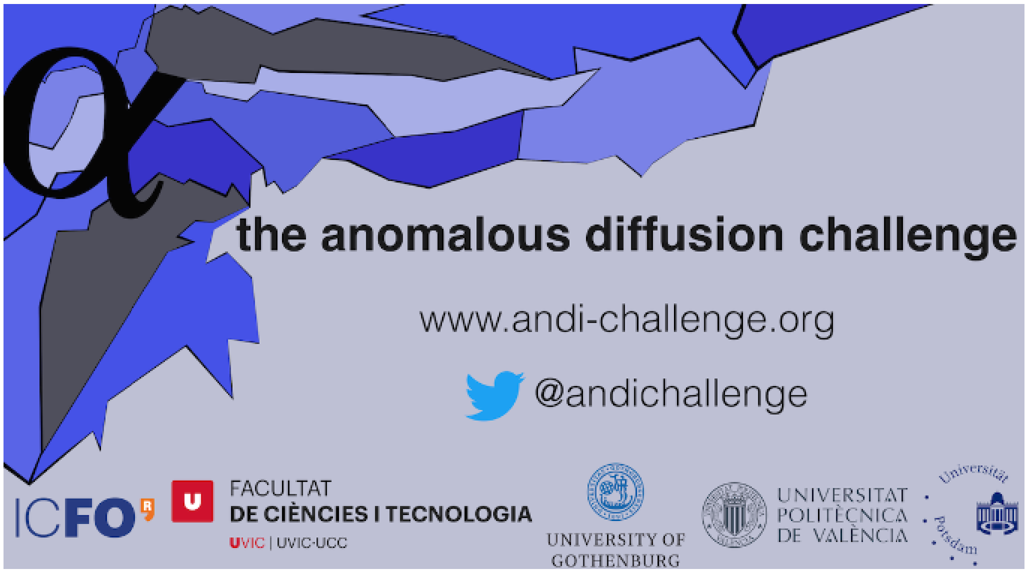

α
the anomalous diffusion challenge
www.andi-challenge.org
@andichallenge
ICFO
FACULTAT DE CIÈNCIES I TECNOLOGIA
UVIC | UVIC-UCC
UNIVERSITY OF GOTHENBURG
UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Universität Potsdam

Fig. 3.3: The AnDi Challenge.

Fotòniques (ICFO), the University of Gothenburg, the Universitat Politèctnica de València and the University of Postdam. Since anomalous diffusion has received an increasing interest from the research community during the last years, due to its convenience to describe an increasing amount of physical scenarios, the main goal of the Challenge was to provide with a benchmark for the current methods to characterize the annomalous diffusing trajectories available in different fields of science.

The AnDi Challenge was publicy available as a competition in the widely known coding competitions website Codalab. Any scientist worldwide was able to enroll the competition and participate. Besides that, the organizers provided with an open-source framework `https://github.com/AnDiChallenge/ANDI_datasets` to create synthetic data sets to be used to develop the solutions to be proposed in the Challenge.

The Challenge consisted of three main tasks:

- Inference of the anomalous diffusion exponent (a regression problem).
- Identification of the underlying diffusion model (a classification problem).
- Segmentation of the trajectories and characterizing the subsequent segments.

We show some examples to ilustrate this three tasks in Figure 3.4. Each of these tasks was addressed in the vectorial spaces of 1, 2 and 3 dimensions, having a total of 9 subtasks. For evaluation purposes an independent data set was used to rank the different participants, the top-rank of them receiving a symbolic award. The main goal of the Challenge is scientific, challenging the scientific community to benchmark their methods and evaluating which of them behave the best, with the aim of establishing a well founded state-of-the-art in the field.

3.1 Data set

Since the procedure to obtain anomalous trajectories in real life is not trivial, the organizers provided with a Python package, and therefore open-source, to allow any participant in the Challenge (or anybody interested in the field) to generate synthetic data sets. This package allowed to build trajectories that follow the 5 underlaying models considered in the Challenge: Continous Random Walk (CTRW), Annealed Transient Time Motion (ATTM), Fractional Brownian Motion (FBM), Lèwy Walk (LW) and Scaled Brownian Motion (SBM), and the anomalous diffusion exponent α

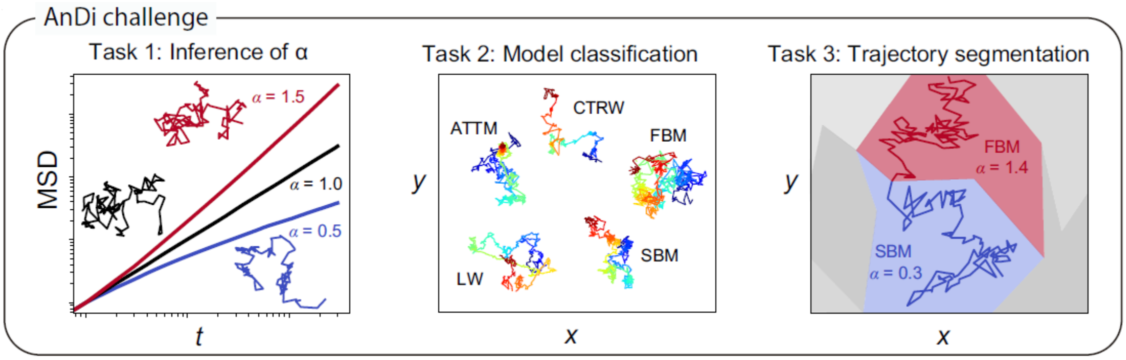


Fig. 3.4: Task 1: inference of the anomalous diffusion expoenent α. We show a normal diffusing trajectory ($\alpha = 1$) in black, a subdiffusive trajectory ($0 < \alpha < 1$) in blue, and a superdiffusive trajectory ($1 < \alpha < 2$) in red. Extracted from [10] with Creative Commons License `https://creativecommons.org/licenses/by/4.0/`

could get values in the range [0.05, 2]. Additionally, in order to resemble experimental data, some noise is added to the trajectories by adding a random number from a gaussian distribution $\mathcal{N}$ with zero mean and standard deviation σ_{noise} to every position in the trajectory. Three different levels of noise were used to corrupt the synthetic trajectories, $\sigma_{noise} \in [0.1, 0.5, 1]$. We then define the signal to noise ratio (SNR) as the inverse of σ_{noise}, resulting in SNR values of [10, 2, 1] respectively. When working with 2 and 3 dimensional trajectories, different levels of noise could be applied to each dimension, being the resulting SNR the average for all directions.

3.1.1 Building trajectories for each underlaying model

We go through the process of simulating the trajectories for each of the five underlaying models considered in the AnDi Challenge, an implementatition of this process was done in a Python package which is avaliable at `https://github.com/AnDiChallenge/ANDI_datasets`.

Continuous Time Random Walk (CTRW). This type of movement is characterized by waiting times between steps which are sampled from a power-law distribution $\psi(t) \approx t^{-\sigma}$ and displacements are sampled from a normal distribution with variance D and zero mean.

Algorithm 1. Generate CTRW trajectory

```
    Input:
        length of the trajectory $T$
        anomalous exponent $\alpha$
        diffusion coeficient D
        Define:
        $\vec{x} \rightarrow$ empty vector
        $\vec{t} \rightarrow$ empty vector
        $\mathcal{N}(\mu, s) \rightarrow$ Gaussian random number generator with mean $\mu$ and
        standard deviation $s$
    $i = 0; \tau = 0$
    While $\tau < T$ do
        $t_i \leftarrow$ sample randomly from $\psi(t) \approx t^{-\sigma}$
        $x_i \leftarrow x_{i-1} + \mathcal{N}(0, \sqrt{D})$
        $\tau \leftarrow \tau + t_i$
        $i \leftarrow i + 1$
    end while
    Return: $\vec{x}, \vec{t}$
```

Annealed Transient Time Motion (ATTM). This movement behaves as a standard Brownian motion, but its diffusion coefficient changes with time. That is, for a random time t_1, the movement follows a Brownian motion with diffusion coeficient D_1, then assumes diffusion coefficient D_2 for a time t_2, and so on.

Algorithm 2. Generate ATTM trajectory

```
Input:
    length of the trajectory $T$
    anomalous exponent $\alpha$
    sampling time $\mathit{\Delta} t$
    Define:
    while $\sigma > \gamma$ and $\gamma > \sigma + 1$ do
      $\sigma \leftarrow$ uniform random number $\in (0, 3]$
      $\gamma = \sigma/\alpha$
    end while
    BM$(D, t, \mathit{\Delta} t) \rightarrow$ generates a Brownian motion trajectory of length $t$
    with diffusion coeficient $D$, sampled at time intervals $\mathit{\Delta} t$
    $\vec{x} \rightarrow$ empty vector
while $\tau < T$ do
    $D_i \leftarrow$ sample randomly from $P(D)D^{\sigma-1}$
    $t_i \leftarrow D_i^{-\tau}$
    number of steps $N_i = round(t_i/\mathit{\Delta} t)$
    $i \leftarrow i + N_i + 1$
    $\tau \leftarrow \tau + N_i \mathit{\Delta} t$
end while
Return: $\vec{x}$
```

Fractional Brownian Motion (FBM). In FBM, $x(t)$ is a Gaussian symmetric process and its EA-MSD scales as $\langle x(t)^2 \rangle = 2K_H t^{2H}$, with H being the Hust exponent, that relates to the anomalous diffusion exponent as $H = \alpha/2$. Another way of understanding FBM is from the Langevin equation with fractional Gaussian noise (fGn) with zero mean and power law correlations.

The FBM presents three regimes. One superdiffusive with $1 < \alpha < 2$, thus having values for the Hurst exponent $1/2 < H < 1$ and where the noise is positively correlated. A second regime which is subdiffusive, with $\alpha \in (0, 1)$, hence the Husrt exponent taking values of $0 < H < 1/2$, and the noise being negatively correlated. And finally a third regime when $\alpha = 1$, with $H = 1/2$ and noise being uncorrelated, what makes FBM behave as pure Brownian motion.

The FBM (`https://pypi.org/project/fbm/`) Python package has been used to generate the FBM trajectories, which uses the Davies-Harte [11] and the Hosking [12] methods. Further details from the implementation can be found in the provided references.

Lévy Walk (LW). LW is a particular case of CTRW, with irregular lapses of time between steps. The difference lies in the distribution of displacements, which in the case of LW is not Gaussian. Instead, the times between steps, or flight times, are generated by the distribution $\psi(t) \approx t^{-\sigma-1}$. The displacements are correlated to the flight times since the probability to make a step $\mathit{\Delta} x$ at some time t and land in a new position, where there will happpen a new waiting period until a new event happens, is $\mathit{\Psi}(\mathit{\Delta} x, t) = \frac{1}{2}\delta(|\mathit{\Delta} x|, vt)\psi(t)$, where v is the velocity. The CTRW output consists of two vectors of irregularly sampled positions and times ($\vec{x}$ and $\vec{t}$).

Algorithm 3. Generate LW trajectory

```
Input:
    length of the trajectory $T$
    anomalous exponent $\alpha$
```

```
        **Define:**
        $\vec{x} \rightarrow$ empty vector
        $\vec{t} \rightarrow$ empty vector
        $v \rightarrow$ random number $\in (0, 10]$
    $i = 0$
    **While** $\tau < T$ **do**
        $t_i \leftarrow$ sample randomly from $\psi(t) \approx t^{-\sigma-1}$
        $x_i \leftarrow (-1)^{\tau} v t_i$, where random $\tau$ is 0 or 1 with equal probability
        $\tau \leftarrow \tau + t_i$
        $i \leftarrow i + 1$
    **end while**
    **Return:** $\vec{x}, \vec{t}$
```

Scaled Brownian Motion (SBM). SBM is defined by the Langevin equation, considering a time-dependent diffusivity K(t)

$$\frac{dx(t)}{dt} = \sqrt{2K(t)}\xi(t), \tag{3.1}$$

where $\xi(t)$ is a white Gaussian noise.

Algorithm 4. Generate SBM trajectory

```
    **Input:**
        length of the trajectory $T$
        anomalous exponent $\alpha$
        **Define:**
        $erfcinv(\vec{a}) \rightarrow$ Inverse complementary error function of $\vec{a}$
        U(L) $\rightarrow$ returns L uniform random numbers $\in$ [0,1]
    **Calculate:**
        $\vec{\Delta x} \leftarrow (1^{\alpha}, 2^{\alpha}, \ldots T^{\alpha}) - (0^{\alpha}, \ldots, (T-1)^{\alpha})$
        $\vec{\Delta x} \leftarrow 2\sqrt{2}U(L)\vec{\Delta x}$
        $\vec{x} \leftarrow cumsum(\vec{\Delta x})$
    **Return:** $\vec{x}$
```

3.2 The Challenge: Evaluation

The evaluation of the performance for the participating teams was as follows:

Task 1: inference of the anomalous diffusion exponent α:

The results were evaluated using the mean absolute error (MAE) for all the predictions, which is calculated as follows

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |\alpha_{i,hat} - \alpha_{i,truth}|, \tag{3.2}$$

where N is the number of trajectories, $\alpha_{i,hat}$ is the prediction and $\alpha_{i,truth}$ the real value for the anomalous exponent for each one of the trajectories.

Task 2: classification according to the 5 underlaying models:

In this task the results were evaluated using the micro F_1 score, which is defined as

$$F_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}, \tag{3.3}$$

We introduce the notation TP for True Positive predictions, that is correct predictions, FP for False Positive, incorrect positive predictions for each class, and FN as False negative, missclassified predictions for each class. With that notation clear we can define precision as $\frac{\text{TP}}{\text{TP + FP}}$, or the portion instances for each class that have been correctly predicted. Similarly, we define recall as $\frac{\text{TP}}{\text{TP+FN}}$, which in practice is the portion of instances of one class that the model was able to correctly classify.

Which is equivalent to defining F_1 score as

$$F_1 = \frac{2\text{TP}}{2\text{TP + FP + FN}}, \tag{3.4}$$

Task 3: segmentation of trajectories. In this task the goal was to perform a change point detection in a given trajectory, and inferring α and assigning the underlying model for each of the 2 resulting trajectories. MAE and $F1$ score were used to measure the performance once the trajectory was split, and root mean squared error (RMSE) to evaluate the accuracy in finding the changepoint localization. We define RMSE as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_{i,hat} - t_{i,truth})^2} \tag{3.5}$$

where $t_{i,hat}$ and $t_{i,truth}$ are the calculated and real changepoint localization in the trajectories. In this task the rank was set by using the mean reciprocal rank which is defined as:

$$MRR = \frac{1}{3} * \left(\frac{1}{rank_{\text{MAE}}} + \frac{1}{rank_{\text{F1}}} + \frac{1}{rank_{\text{RMSE}}}\right), \tag{3.6}$$

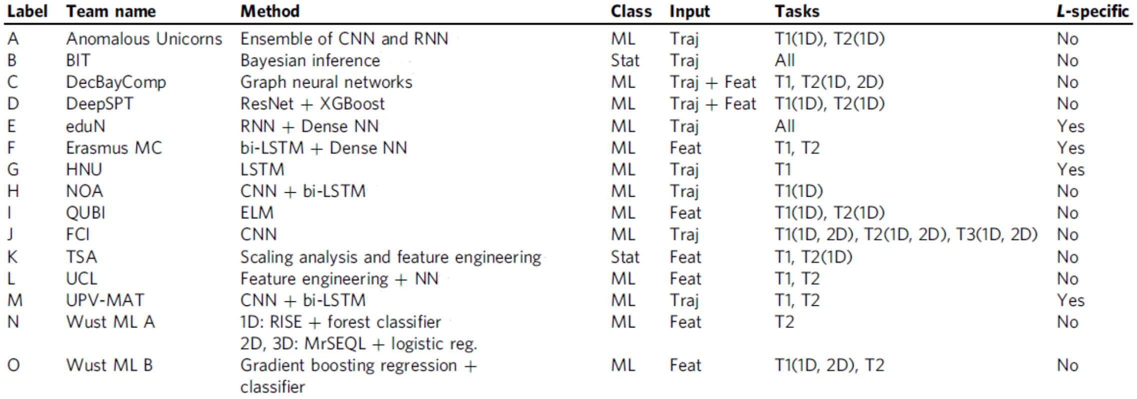
being each of the ranks the position in the ordered list based on the value of each of the metrics. MAE and $F_{1-\text{score}}$ are calculated independently for each segment and results are averaged over them.

3.3 The Challenge: Methods

A team could participate in any or all of the 9 subtasks during the Challenge. The participation by task was as follows: 13 teams participated in task 1, 14 in task 2 and 3 in task 3. Each task was proposed for trajectories in one, two or three dimensions, resulting in the 9 subtasks aforementinoned. In Figure 3.5 we show a relation of the participants, including the following information:

- Label: label assigned to each team in the plots.
- Team name: Name selected by the team for the competition.
- Method: a brief description of the method used by each team.
- Class: either if the team used machine learning or classical statistics approach.
- Input: input to the method, being raw trajectories (Traj) and/or features extracted from the trajectories (Feat).
- Tasks: tasks in which every team participated (if necessary, also subtasks are indicated).
- L-specific: indicates if the proposed method is trajectory lenght specific or not.

As we can see in Figure 3.5, 13 out of 15 teams opted by using a variety of machine learning methods, which included recurrent neural networks (RNN), convolutional neural networks (CNN), a combination of these two, graph neural networks, extreme learning machines, sequnece learners (transformers) and gradient boosting machines. While other methods used statistical approaches like Bayesian inference, random interval spectral ensemble and temporal scaling. In this same Figure



Label	Team name	Method	Class	Input	Tasks	*L*-specific
A	Anomalous Unicorns	Ensemble of CNN and RNN	ML	Traj	T1(1D), T2(1D)	No
B	BIT	Bayesian inference	Stat	Traj	All	No
C	DecBayComp	Graph neural networks	ML	Traj + Feat	T1, T2(1D, 2D)	No
D	DeepSPT	ResNet + XGBoost	ML	Traj + Feat	T1(1D), T2(1D)	No
E	eduN	RNN + Dense NN	ML	Traj	All	Yes
F	Erasmus MC	bi-LSTM + Dense NN	ML	Feat	T1, T2	Yes
G	HNU	LSTM	ML	Traj	T1	Yes
H	NOA	CNN + bi-LSTM	ML	Traj	T1(1D)	No
I	QUBI	ELM	ML	Feat	T1(1D), T2(1D)	No
J	FCI	CNN	ML	Traj	T1(1D, 2D), T2(1D, 2D), T3(1D, 2D)	No
K	TSA	Scaling analysis and feature engineering	Stat	Feat	T1, T2(1D)	No
L	UCL	Feature engineering + NN	ML	Feat	T1, T2	No
M	UPV-MAT	CNN + bi-LSTM	ML	Traj	T1, T2	Yes
N	Wust ML A	1D: RISE + forest classifier 2D, 3D: MrSEQL + logistic reg.	ML	Feat	T2	No
O	Wust ML B	Gradient boosting regression + classifier	ML	Feat	T1(1D, 2D), T2	No

Fig. 3.5: Participants and their methods in the AnDi Challenge.

we can see what kind of input the teams used. With 7 teams using the raw trajectories, 6 using engineered features, and 2 using a combination of both. Finally, we can group the teams taking into account if the method they used was length specific (that is, they require specific training for different trajectory lengths) or not, with just 4 teams presenting length specific methods.

We next give further details in the different teams' methods:

Team A used an ensemble of CNN and RNN. For Task 2, they used CNN and RNN in parallel as features extractor whose output was delivered to 2 fully connected layers which predicted the class. For regression they used the same architecture but trained specific regressors per each underlaying model. So, they firstly classify the trajectory by underlaying method and then infer the α exponent. More detailed information and code can be found in `https://github.com/BorjaRequena/AnDi-unicorns`.

Team B presented a solution based on Bayesian inferences with a uniform prior for the 5 underlaying models for Task 2, and 5 priors, one per underlaying models, according to the α ranges for each underlaying model. Same approximation was done for Task 3. Additional information can be found in `https://github.com/mlomholt/andi`.

Team C built a solution based on graph neural metworks. First they build a graph from each trajectory, where each position is a node in the graph, and edges are drawn between nodes based on their time difference. At this point, additional information is added to nodes from features calculated from normalized positions such as cumulated distance covered from origin, maximal step size from origin, etc. Then, these graphs were used to train a graph neural network (convolution) used as an encoder whose output is then forwarded to specialized modules for the different tasks (regression and classification). This solution accepts inputs of any length, which are encoded into fixed length vectors in a latent space. For more information on this method please refer to [13] and `https://github.com/DecBayComp/gratin`.

Team D used a combination of ResNet (deep CNN) and XGBoost. For the classification task they split the data in 7 different classes, splitting FBM and SBM in two subclasses each, one subdiffusive and one superdeiffusive, in order to better distinguish between those 2 different behaviours inside one single (two, in fact) underlaying model. On the one hand they fed the ResNet with portions of lenghth 100 from each trajectory, and XGBoost with up to 20 features they engineered, merging both branches with fully connected branches to produce the model's output (either inference of α or classification). Further information about this method can be found in `https://github.com/TaegeunSONG/DeepSPT`.

Team E proposed a method based on long short term memory (LSTM) RNN. They stacked 2 LSTM layers with dimensions 250 and 50, followed by 2 fully connected layers, a first one with 20 nodes and a second one with one node for Task 1 (inference of α) and 5 nodes for Task 2, one per class. They trained up to 6 different models with independent data of different lengths for the classification task (lenghts 25, 65, 125, 225, 425 and 825). And then, given a trajectory, they used the

2 networks with lengths closer (up and down) to the trajectory length to provide with the prediction. A similar approach was used for the inference task, but using 14 different networks with trajectory lengths 25, 50, 65, 75, 125, 165, 225, 325, 425, 525, 625, 725, 825 an 925. Detailed information about this method can be found in [14] and `https://github.com/booste/andi_for_organizers`.

Team F extracted 6 features that vary with the dimension (x, y or z axis displacements, distances, mean of distances and/or angles) at each trajectory point and these features fed stacked bidirectional LSTMs followed by 1 (regression) or 5 (classification) neuron/s fully connected layer to produce the prediction. More detailed information about this method can be found in `https://github.com/hkabbech/FEST_AnDiChallenge`.

Team G participated in Task 1 with a model based on LSTM. They trained 43 different models for 43 trajectory length ranges in 1 dimension. For trajectories with dimension 2 and 3 they applied the 1 D models to each trajectory axis and computed the average of the α predicted for each axis, resulting in their final prediction. For more detailed information about the method see `https://github.com/huangzih/AnDi-Challenge`.

Team H presented a solution for task and 1 D trajectories based on a mixture of 2 CNN used as feature extractors whose output fed 3 stacked bidirectional LSTM layers. More details can be found in `https://github.com/NicoFirbas/ConvLSTM_AnDI`.

Team I joined task 1 for 1 dimensional trajectories and opted for a solution based in extreme learning machine from extracted features such as absolute displacement, logarithmic mean displacement, logarithmic mean squared displacement and correlation of displacement. Further details can be seen in [15] and `https://github.com/qubilab/AnDi_ELM`.

Team J proposed a method for all tasks and dimensions 1 and 2 based on up to 4 blocks of CNN, each of them consisting of 4 stacked convolutional layers followed each one of them by a batch normalization layer. To obtain the prediction the model used 3 fully connected layer of size 512, 256 and 1 for the regression task, or 5 for the classification task. For more information about this method see `https://github.com/tsmbland/andi_challenge`.

Team K participated in task 1 and 2 for 1 dimensional trajectories performing the statistical method scaling analysis over computed features. More details can be seen in `https://github.com/ErezAgh/ANDI-challange-codes-`.

Team L combined feature engineering based on classical statistics (momenta and statistics at different combinations of steps, displacement relative change, normalized power spectral density statistiscs and others) with fully connected layers. They implement the machine learning method in 3 different steps: in the first one they consider 2 classes, LW and the rest of underlaying models. Then they look for ATTM trajectories and finally classify the remaining between the rest of classes. Detailed information about the method can be found in [16] and `https://github.com/sam-labUCL/CONDOR`.

Team M was our team. We used 2 CNN layers to extract features from the trajectories and 3 bidirectional LSTMs followed by 2 fully connected layers to get the prediction. Our method has beed discussed in the previous chapter in this Thesis, but more information can be seen in [17] and in `https://github.com/OscarGariboiOrts/ANDI_Challenge`.

Team N participated only in the task 2. They performed feature engineering and used statistical methods to classify trajectories within the 5 underlaying models. For 1 diemnsional trajectories they used the Random Interval Spectral Ensemble (RISE), that uses several feature extraction transformers like fitted auto-regressive coefficients, power spectrum coefficients and estimated correlation coefficients. For 2 and 3 dimensional trajectories they used Mr-SEQL, a univariate time series classifier which trains logistic regressions with features extracted from several symbolic representations of trajectories. More infromation about this method can be found in `https://github.com/szwabin/ANDI-challenge/`.

Team O used gradient boosting method (xgboost from Python's sklearn) using extracted features from the trayectories using statistics. They used features like fractal dimension, Gaussianity, MSD ratio, normalized maximum excursion, among other. For further details on this method see [18] and `https://github.com/HannaLochOlszewska/ANDI_challenge`.

3.4 Our results

We participated in the Challenge in the first two tasks, using a combination of convolutional and recurrent neural networks, whose architecture was defined in the previous Chapter, ConvLSTM. For Task 1 we trained 12 different models for each subtask, being each model specialized in a fraction of the possible trajectories lengths. Thats is, we trained one model for each of the following trajectories' length ranges: [10, 20], [21, 30], [31, 40], [41, 50], [51, 100], [101, 200], [201, 300], [301, 400], [401, 500], [501, 600], [601, 800], [801, 1000]. We used different amount of trajectories to train each specific model, depending on the memory required to process them, adjusting to our equipment limitations. The range of trajecotries used to train the models varies from $8,000,000$ for shorter trajectories and 1 dimension, to $2,000,000$ for larger trajectories and 3 dimensions. And for Task 2 we used one single model for each subtask, capable to deal with trajectories in the whole range considered in the Challenge (i.e. [10, 1000]). In this case we used $2,000,000$ trajectories to train all models (1 per subtask). 10% of the data available for training was used for validation at each epoch. It is worth mentioning that the complete training process for Task 1 and 2 took us 3 months, due to the aforementioned limititation inherent to our equipment. The Challenge organizers created an on-line tool to measure the performance of each of the participating team `htp://andi-challenge.org/interactive-tool/`) where they show how every team's predictions rate, being able to split by trajectory length, α, underlaying diffusion model and dimension, for Task 1 and Task 2. Figure 3.6 shows our team's model performance for Task 1 and Task 2 in the 1 dimensional trajectories case.

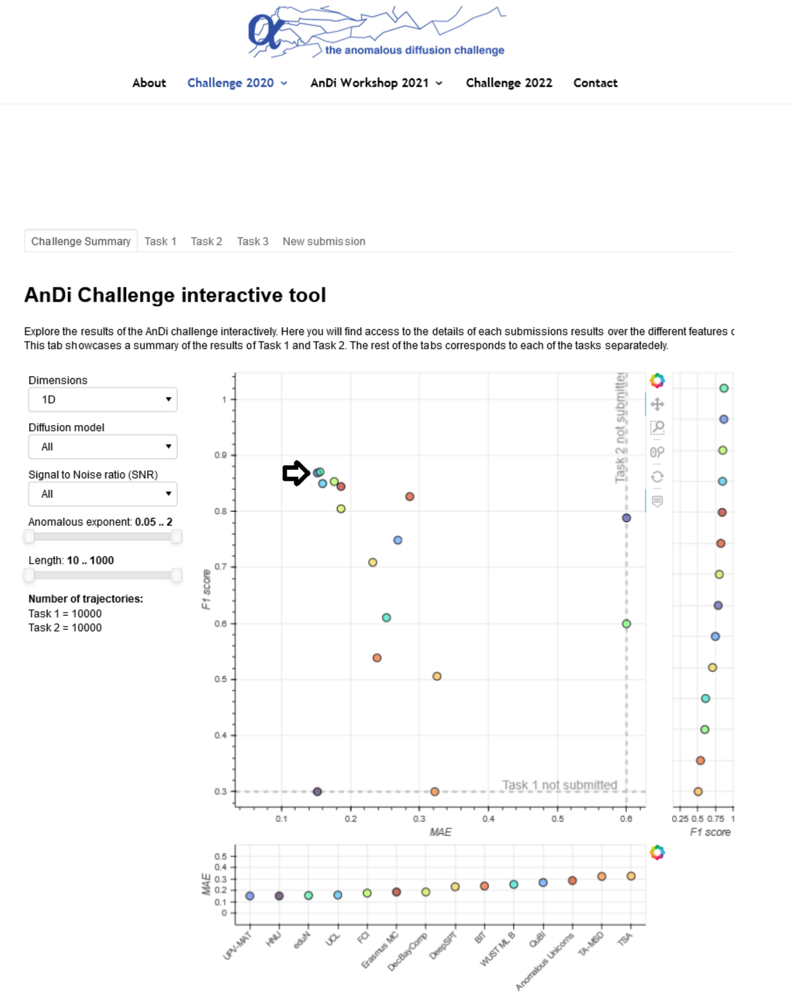
Our method was awarded with the first position in the Task 1, subtask 1 dimension, and ranked in positions 2^{nd}, 4^{th} and 3^{rd} in the subtasks related to Task2. In Figure 3.6 we present the performance of our model in the case of 1 dimensional trajectories, both for Task 1 (bottom) and Task 2 (up). For Task 2 results, our result has been pointed by an arrow, since in order to ease the identification. It is easy to see how close the 4th best performing teams were one to the other, probably indicating that we have reached the top possible performance of the current machine learning methods.

3.5 Corollary

As a result of the good performance we were invited to present our job in the AnDi Workshop (`http://andi-challenge.org/workshop/`), where we received our award 3.7, gave a talk explaining our method and presented a poster of a research project that resulted in our second paper in this Thesis, *Gramian Angular Fields for leveraging pre-trained computer vision models with anomalous diffusion trajectories.*

The participation in the AnDi challenge was the embryo of this Thesis. We dived into the anomalous diffusion problem and realized how machine learning methods can help to shed light to some problems in the field of physics and mathematics. The celebration of the AnDi Workshop was intented to happen during December 2020, but the event was postponed due to Covid-19 pandemic restrictions of mobility and gathering. We were willing to attend the Workshop to know the rest of participant, since we developed a very interesting and amazing competition. But, as usual in life, Covid-19 also opened some new doors to the usage of machine learning, as we show in the next chapter of this Thesis.

Additionally, I supervised the work of a master student who also took part in the AnDi Challenge (Team H). His work evolutioned to the implemetation of a transformer which can be used to infer the anomalous exponent α and to classify the underlaying model for one dimensional trajectories. I co-authored the paper [19] which is currently under revision for publishing in *Journal of Physics E.* This paper has not been included in this Thesis since the main author is Nicolás Firbas (the student) but I contributed with supervision and advicing.



AnDi Challenge interactive tool

Explore the results of the AnDi challenge interactively. Here you will find access to the details of each submissions results over the different features c
This tab showcases a summary of the results of Task 1 and Task 2. The rest of the tabs corresponds to each of the tasks separatedely.

Dimensions
1D ▾

Diffusion model
All ▾

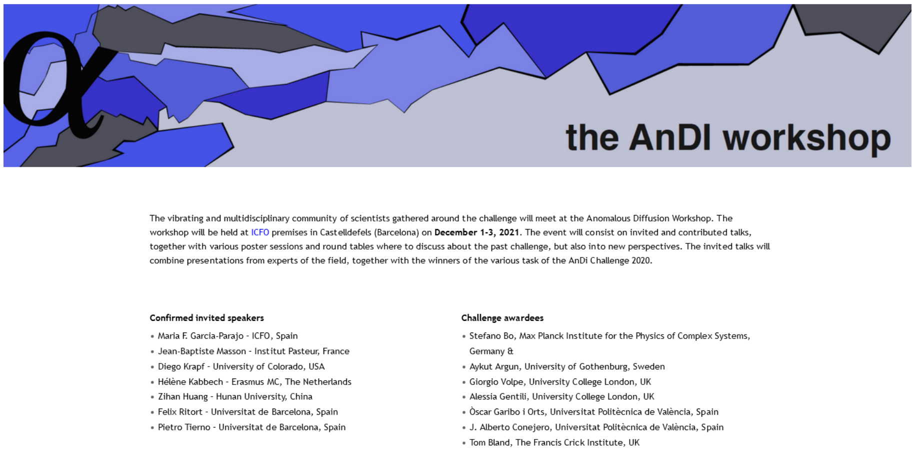
Signal to Noise ratio (SNR)
All ▾

Anomalous exponent: **0.05 .. 2**

Length: **10 .. 1000**

Number of trajectories:
Task 1 = 10000
Task 2 = 10000

Fig. 3.6: The AnDi Workshop.



The vibrating and multidisciplinary community of scientists gathered around the challenge will meet at the Anomalous Diffusion Workshop. The workshop will be held at ICFO premises in Castelldefels (Barcelona) on **December 1-3, 2021**. The event will consist on invited and contributed talks, together with various poster sessions and round tables where to discuss about the past challenge, but also into new perspectives. The invited talks will combine presentations from experts of the field, together with the winners of the various task of the AnDi Challenge 2020.

Confirmed invited speakers

- Maria F. Garcia-Parajo - ICFO, Spain
- Jean-Baptiste Masson - Institut Pasteur, France
- Diego Krapf - University of Colorado, USA
- Hélène Kabbech - Erasmus MC, The Netherlands
- Zihan Huang - Hunan University, China
- Felix Ritort - Universitat de Barcelona, Spain
- Pietro Tierno - Universitat de Barcelona, Spain

Challenge awardees

- Stefano Bo, Max Planck Institute for the Physics of Complex Systems, Germany &
- Aykut Argun, University of Gothenburg, Sweden
- Giorgio Volpe, University College London, UK
- Alessia Gentili, University College London, UK
- Òscar Garibo i Orts, Universitat Politècnica de València, Spain
- J. Alberto Conejero, Universitat Politècnica de València, Spain
- Tom Bland, The Francis Crick Institute, UK

Fig. 3.7: The AnDi Workshop.

4 ValenciaIA4Covid

I've got a theory that if you give 100% all of the time,
somehow things will work out in the end.

Larry Bird, GOAT.

The Covid-19 pandemic came abruptly for everyone of us, in every country, for any professional job. In front of it, scientists found ourselves in front of a crossroads: try to use our knowledge to help or stay away from front edge, to avoid introducing more noise. No one had a good answer on what to do, and most scientists did what they thought was the best. In our case, we had in our hands a tool that was extremely successful in the AnDi Challenge. On top of this, we had collaboration with worldwide recognized scientist like Nuria Oliver. The XPrize challenge came to our knowledge and we found that participating again in a collaboration effort was a reasonable way of contributing, within our strengths.

To put things in context, from Wikipedia we know that "XPrize Foundation is a non-profit organization that designs and hosts public competitions intended to encourage technological development to benefit humanity. The XPRIZE mission is to bring about radical breakthroughs for the benefit of humanity through incentivized competition. It fosters high-profile competitions to motivate individuals, companies and organizations across all disciplines to develop innovative ideas and technologies that help solve the world's grand challenges."

The XPrize foundation has been setting prizes from 1996, with the common characteristic to all prizes that they would somehow help to develop a better society. We can check Wikipedia (`https://en.wikipedia.org/wiki/X_Prize_Foundation#Prizes_and_events_overseen`) for a complete list, but to mention some of them:

- Wendy Schmidt Oil Cleanup XPrize (2010-2011) wanted to inspire innovative solutions that will speed the pace of cleaning up seawater surface oil resulting from spillage from ocean platforms, etc.
- Water Abundance XPrize (2016-2018) focused in addressing water scarcity in the developing world.
- The Global Learning XPrize (2014-2019) pushed the creration of mobile apps to improve reading, writing and arithmetic in developing nations.

In October 30th the XPrize foundation started the early registration for the XPrize Pandemic Response Challenge (see Figure 4.1), where teams should face two challenges in one. First, the teams should build a prediction model that was able to predict up to 90 days in the future the daily reported Covid-19 cases in 236 countries and regions world wide. Second, the teams should create a Non-Pharmaceutical Interventions (NPI) prescriptor, as a tool for the decision makers to ponder the trade-off between new Covid-19 daily cases and social cost for the NPI implementations. Where NPI were non-pharmaceutical interventions which decission makers (a.k.a. goverments) could decide to implement or not, such as school closing, lockdowns, etc. In opposition to the normal extension of

XPRIZE
WHO
WHAT
ACT
STORIES
PANDEMIC RESPONSE CHALLENGE
PHASE Awarded
XPRIZE PANDEMIC RESPONSE CHALLENGE
Cognizant

Fig. 4.1: Pandemic Response Challenge from XPrize.

XPrize challenges, wich usually extend over several years, the Pandemic Response Challenge was designed to last just for 3 months. Obviuosly, if the aim of the competition was to build models to help decision makers during the pandemic, the whole process had to be fast.

The following content is extracted from the paper I co-author *Open Data Science to fight COVID-19: Winning the 500k XPRIZE Pandemic Response Challenge* [20], which was presented during the 2021 edition of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD). I emphasize here that our paper was awarded as the Best Applied Data Science Paper Award `https://2021.ecmlpkdd.org/index.html@p=2148.html`.

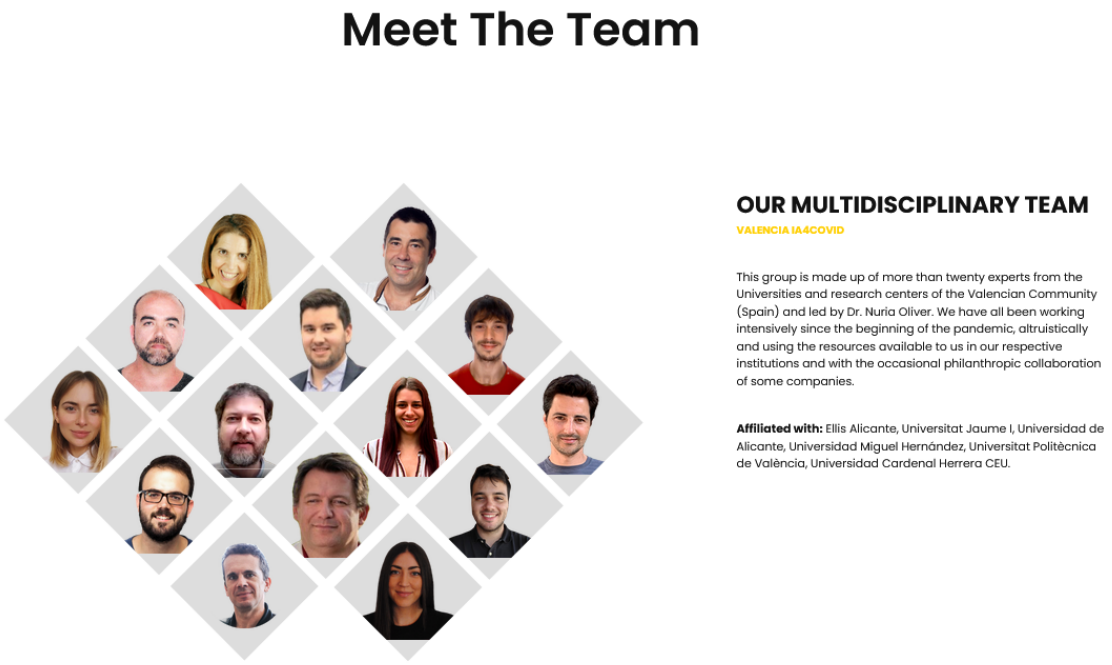
Data for the challenge consisted of time series data where we had to consider not just the reported daily cases from every country or region, but the applied NPIs established in each of them. The XPrize organization provided with a baseline model and due to our expertise working with time series data and LSTM recurrent neural networks (an expertise built in previous research and sharpened in the AnDi Challenge), we decided to look for mates to join this new adventure.

Since March 2020, the Valencian Goverment worked together with a group of local scientists from universities in the territory. This group was so called Data Science for Covid-19 Group and helped the local goverment by implementing SIR models to identify how the pandemic spread, such work can be found in [20–26]. This group was the embryo for the ValenciaIA4Covid team (see Figure 4.2) that registered into the challenge just a couple of days before the deadline, after having implemented a functional predictor.

4.1 Data

Historically there has not been data available in a daily basis for most countries worldwide, but during Covid-19 global pandemic extensive data was captured and openly shared for the general population and for the scientific community in particular. During the XPrize Pandemic Response Challenge we could use the Oxford COVID-19 Government Response Tracker set[1] as the main data source. In this data repository we could find daily reported Covid-19 cases (that is, *case*-related data), and NPIs being implemented (*action or NPI*-related data) in 186 countries, the 50 US states and the 4 countries in the United Kingdom, having a grand total of 236 countries or regions, which we denoted as GEO. The *action or NPI*-related data included the level of activation for each NPI for each GEO and day. In Table 4.1 we show the 12 NPIs we cosidered during the Challenge, which we split in two main groups: *confinement-based* and *public health-based*. For a more detailed view of NPIs and the meaning of the different levels of activation, please see Table 4.2 for confinement NPIs and Table 4.3 for public health NPIs.

[1] `https://www.bsg.ox.ac.uk/research/research-projects/coronavirus-government-response-tracker`



Meet The Team

OUR MULTIDISCIPLINARY TEAM

VALENCIA IA4COVID

This group is made up of more than twenty experts from the Universities and research centers of the Valencian Community (Spain) and led by Dr. Nuria Oliver. We have all been working intensively since the beginning of the pandemic, altruistically and using the resources available to us in our respective institutions and with the occasional philanthropic collaboration of some companies.

Affiliated with: Ellis Alicante, Universitat Jaume I, Universidad de Alicante, Universidad Miguel Hernández, Universitat Politècnica de València, Universidad Cardenal Herrera CEU.

Fig. 4.2: ValenciaIA4Covid team.

Table 4.1: NPIs considered in the Challenge and their possible activation values.

NPI name	**Values**	**NPI name**	**Values**
C1. School closing	[0,1,2,3]	C7. Internal movement restrictions	[0,1,2]
C2. Workplace closing	[0,1,2,3]	C8. International travel controls	[0,1,2,3]
C3. Cancel public events	[0,1,2]	H1. Public information campaigns	[0,1,2]
C4. Restrictions on gatherings	[0,1,2,3]	H2. Testing policy	[0,1,2,3]
C5. Close public transport	[0,1,2]	H3. Contact tracing	[0,1,2]
C6. Stay at home requirements	[0,1,2,3]	H6. Facial coverings	[0,1,2,3,4]

4.2 Predictors of COVID-19 cases

As stated before, the main goal of the first part of the Challenge was to build a predictor for 236 GEOs and for up to 180 days in the future. We will be using the next notation:
1. Population (P^j)**:** P^j is the reported population for GEO j. We consider each GEO's population being constant.
3. NewCases (X_n^j)**:** Reported daily number of new cases on day n and GEO j, is denoted by X_n^j. We consider March 11th as the first day in our temporal data.
4. ConfirmedCases (Y_n^j)**:** Cumulative confirmed cases until day n in GEO j and denoted by $Y_n^j = \sum_{i=1}^n X_i^j$.
5. SmoothedNewCases (Z_n^j)**:** Averaged number of new cases between days $n-K+1$ and n in GEO j as $Z_n^j = \frac{1}{K}\sum_{i=0}^{K-1} X_{n-i}^j$. We implemented this measure since different GEOs had different imputation polices (some report cases on weekends, some not; some reports in a daily basis, some 3 days per week). We decided to use the week as the smoothing unit, so we set $K = 7$.
6. CaseRatio (C_n^j)**:** Cases ratio for two consecutive days is computed as $C_n^j = Z_n^j / Z_{n-1}^j$. It shows if the number of cases is growing or decreasing in a daily basis.
7. Susceptible Population (S_n^j)**:** Number of individuals in a given population who are keen to be infected with coronavirus on day n and for GEO j.
8. ScaledCaseRatio (R_n^j)**:** CaseRatio C_n^j harmonized by the proportion of susceptible individu-

Table 4.2: Confinement Intervention Policies and the meaning for each value.

NPI name	**Level 0**	**Level 1**	**Level 2**	**Level 3**	**Level 4**
C1. School closing	Nothing	Recommend closing	Partial closing	Complete closing	
C2. Workplace closing	Nothing	Recommend closing (or work from home)	Require closing (or work from home for some sectors/categories of workers)	Require closing (or work from home all-but-esential workplaces)	
C3. Cancel public events	Nothing	Recommend cancelling	Require cancelling		
C4. Restrictions on gatherings	Nothing	Cancel very large gatherings (above 1000 people)	Cancel gatherings between 100-1000 people	Cancel gatherings between 11-100 people	Cancel gatherings of 10 or less people
C5. Close public transport	Nothing	Recommend closing (or significantly reduce volume, routes and/or means of transport available)	Require closing (or prohibit most citizens from using it)		
C6. Stay at home requirements	Nothing	Recommend not leaving home	Require not leaving home with exceptions for daily exercise, grocery shopping and essential trips	Require not leaving home with minimal exceptions (e.g. allowed to leave once a week, or only one person can leave at a time, etc)	
C7. Restrictions on internal movement	Nothing	Recommend not to travel between regions/cities	Internal movement restrictions in place		
C8. Internaltional travel controls	Nothing	Screening arrivals	Quarantine arrivals from some or all regions	Banning on arrivals froms some regions	

als in GEO j, $R_n^j = C_n^j \frac{P^j}{S_n^j}$. Since it depends on the propotion of susceptible individuals, it captures the effects of a finite population.
9. Action (A_n^j): Applied NPIs in GEO j on day n.

Predictions (or estimations) are denoted by the use of $\widehat{}$ symbol, that is, $\widehat{X}_n^j$ is the estimated new cases and $\widehat{R}_n^j$ the estimated scaled case ratio, both for GEO j and day n.

4.2.1 Baseline predictor.

The Challenge organization provided with a baseline model which we show in Figure 4.3. This model consists of 2 Long Short Term Memory (LSTM) based branches, one for processing daily cases (context branch) and the other in charge of the NPIs (action branch), with a lambda merge layer to combine both.

We feed the context LSTM h with the values of R_n in the previous T days in GEO j, or $\mathbf{R}_{n-1}^j = (R_{n-T}^j, \ldots, R_{n-1}^j)$, while the action branch LSTM g is fed with the NPIs 12-dimensional matrix with NPIs applied in the previous T days in GEO j, or $\mathbf{A}_{n-1}^j = (A_{n-T}^j, \ldots, A_{n-1}^j)$.

In order to mitigate the noise induced by the different Covid-19 cases reporting policies between GEOs we set $T = 21$ (as in [27]), that is, three weeks. Having this value also permits the model to

Table 4.3: Public Health Intervention Policies and the meaning for each value.

NPI name	**Level 0**	**Level 1**	**Level 2**	**Level 3**	**Level 4**
H1. Public information campaigns	No Covid-19 public information campaigns	Public officials urging caution about Covid-19	Coordinated public information campaign (e.g. across traditional and social media		
H2. Testing policy	No testing policy	Only those who both (a) have symptoms AND (b) meet specific criteria (e.g. workers, admitted to hospital, came into contact with a known case, returned from overseas)	Testing of everyone showing Covid-19 symptoms	Open public testing (e.g. drive through testing available to asymptomatic people)	
H3. Contact tracing	No contact tracing	Limited contact tracing; not done for all cases	Comprehensive contact tracing; done for all identified cases		
H6. Facial coverings	No policy	Recommended	Required in some specified shared or public spaces outside the home with other people present, or some situations when social distancing is not possible	Required in all shared or public spaces outside the home with other people present or all situations when social distancing is not possible	Required outside the home at all times regardless of location or presence of other people

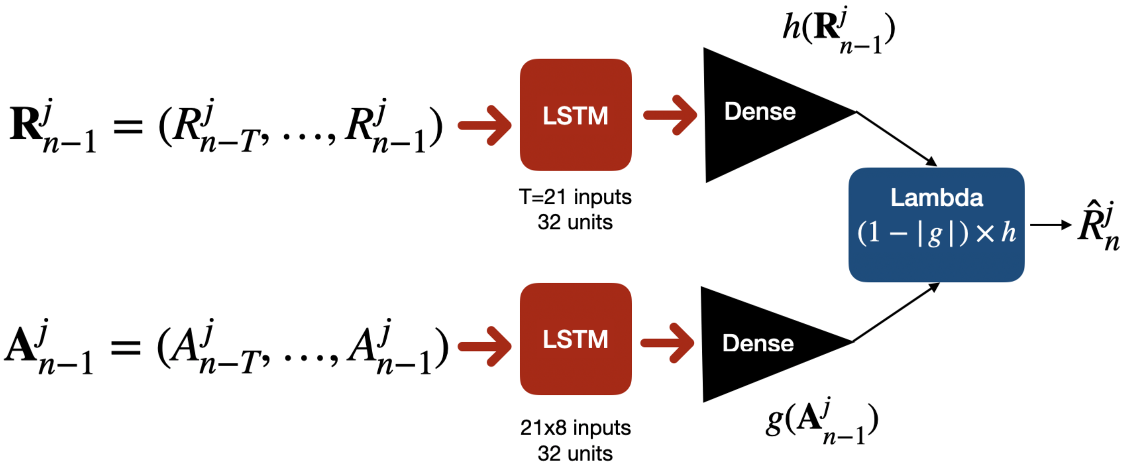


Fig. 4.3: ValenciaIA4Covid model.

observe the whole incubation period for Covid-19, which is agreed to be of 12-15 days from being exposed to the virus to being tested and confirmed [28].

The lambda layer for day n is the predicted $\widehat{R}^j_n$ given by

$$\widehat{R}^j_n = f(\mathbf{A}^j_{n-1}, \mathbf{R}^j_{n-1}) = (1 - g(\mathbf{A}^j_{n-1}))h(\mathbf{R}^j_{n-1}), \tag{4.1}$$

with $g(\mathbf{A}^j_{n-1}) \in [0, 1]$ and $h(\mathbf{R}^j_{n-1}) \geq 0$. See [27] for in-detail description of the baseline model. It is worth mentioning that when making predictions into the future, the R^j_{n-i} values in the vector $\mathbf{R}^j_n$ are replaced by the estimations provided by the predictor, or $\widehat{R}^j_{n-i}$, for $n - i >$ current_day, $i = 1, \dots, T$.

4.2.2 ValenciaIA4COVID (V4C) predictor

Starting from the baseline architecture, our model consisted of LSTM-based branches: a *context* branch, to model the R_n time series and an *action* branch, where the time series of Non-pharmaceutical Interventions were modeled. Although we did not consider public health-based NPIs, we included some major modifications to the baseline predictor which lead us to our ValenciaIA4COVID or V4C predictor. We get into details for the modifications to each of the branches.

4.2.2.1 Context branch

Since the confirmed cases time series were very divergent for the different GEOs, one single LSTM network resulted in its weight matrices being full rank. To deal with the variability inherent to the difference between GEOs (stationality differences between north and south, population density, etc) we decided to build a bank of LSTM for the context branch, as it can be seen in Figure 4.4(up).

Bank of context models. We wanted to find sets of GEOs with similar behaviour with respect to the number of confirmed cases, and to do so we clustered the GEOs using the K-Means algorithm applied to the time series of reported number of COVID-19 cases per 100K inhabitants. In Figure 4.5 (left) we show the result of applying K-means algorith from 1 to 40 clusters. In the x axis we represent the number of desired clusters k while in the y axis we have the summation of the squared distances from each point in each cluster to its centroid. We set the optimal number of clusters being 15 since the squared distance reaches an stability point, having a good trade-off between squared root distance (precision) and number of clusters (computational cost). Also in Figure 4.4 (right) we plot the distribution of the number of cases per day for the GEOs in one of the 15 resulting cluster. This cluster contains the GEOs Colombia, Mexico, South Africa, Florida and Texas.

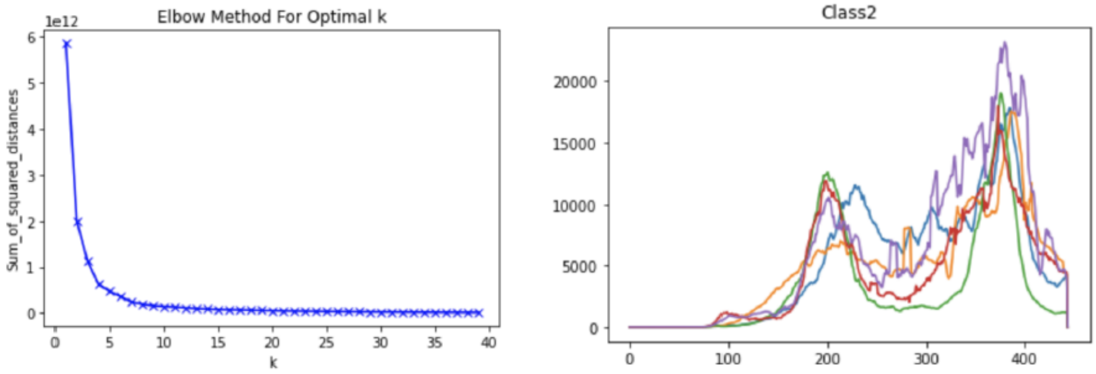


Fig. 4.4: Results of applying the Elbow Method to the K-Means clustering (left) and example of the GEOs in one of the resulting clusters (right).

With the aim of assuring our clustering was correct we performed a t-distributed neighbor embedding (t-SNE) analysis. t-SNE is a statistical method that models each high-dimensional point by a 2 or 3 dimensional point. Similar high dimentional points in the 2 o 3 dimensional vector space are projected in the new 2 or 3 dimensional space in such a form that the resulting points are close to each other, and dissimilar points in the high-dimensional space result in distant point in the projected vector space, both with high probability.

t-SNE consists of 2 main stages:

1. Given a set of N elements from the high-dimensional space, t-SNE computes the probabilities p_{ij}, which are proportional to the similarity of each pair x_i and x_j as
$$p_{j|i} = \frac{exp(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} exp(-\left\|x_i - x_k\right\|^2 / 2\sigma_i^2)}, \tag{4.2}$$
where, by definition $p_{i|i} = 0$, and $\sum_j p_{j|i} = 1$ for all i. We then define
$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \tag{4.3}$$
where $p_{ij} = p_{ji}$, $p_{ii} = 0$ and $\sum_{i,j} p_{i,j} = 1$
2. t-SNE searchs for a 2 o 3 dimensional map $y_1, \ldots, y_N$ that preserves the similarities previously computed. To do this, it computes similarities between points in the projected space y_i and y_j with a similar method. For every pair (i,j) where $i \neq j$, a distance metric q_{ij} is calculated as follows
$$q_{ij} = \frac{(1 + \left\|y_i - y_j\right\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + \left\|y_k - y_l\right\|^2)^{-1}}. \tag{4.4}$$
The position of the points y in the 2 or 3 dimensional space vector are calculated by minimizing the Kullback-Leibler divergence of the distribution P form the distribution Q, which writes as
$$KL(P\|Q) = \sum_{i \neq j} p_{ij} log \frac{p_{ij}}{q_{ij}}, \tag{4.5}$$
where the minimization of the Kullback-Leibler divergence from the y_i points is done using gradient descent. Obtaining a map that preserves the similarities between the high-dimensional points.

In Figure 4.5 we depict the distribution of the GEOs according to the tSNE representation of the clusters they are assigned to.

Once the GEOs were clustered, we trained a LSTM model with data from the 20 most-affected GEOs, which we call *reference*, and 15 different LSTM models using data from all the GEOs in each of the 15 clusters, which we call *cluster*. We used March 11th, 2020 as the starting date for training the models. After training we evaluated the reference and all the cluster models on testing data for all the GEOs. We set two different periods for the testing, a short-term evaluation (from December 1st to December 21st) and a long-term evaluation (from from November 1st to to December 21st) in 2020. In order to minimize the number of models in our bank (we wanted less than 15 models for the sake of avoiding overfitting) we selected the model with the lowest MAE per 100K inhabitants in each GEO, applying Occam's razor principle. With the aim of lowering the number of clusters we compared the performance of the *reference* model with the best performing cluster model, assigning a GEO to the *reference model* when similar performance was achieved, ending with a bank of nine models: the *reference* model, applied in 135 GEOs; and eight *cluster* models applied in the remaining GEOs. In Figure 4.6 we show the result of the clustering of GEOs using 15 clusters, with the geographical distribution of each cluster. And in Figure 4.7 we show the final predictor model assigned to each GEO.

A visualization of the cluster and model assignments can be found here[2].

[2] `https://tinyurl.com/cjstz4yc`

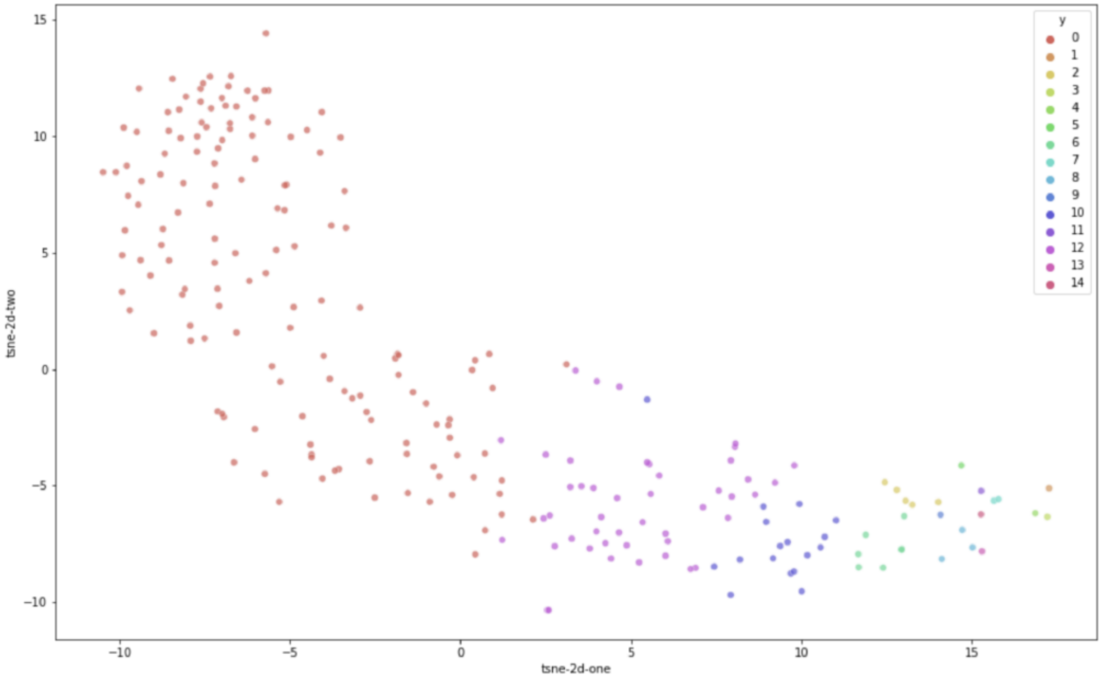


Fig. 4.5: tSNE spatial distribution of the clusters.

LSTMs Architecture. We implemented two different LSTM-based architectures, as shown in Figure 4.8 (right): one for the *reference* model and the other for each of the eight *cluster* models. The *reference* model includes a convolutional layer with ReLu activation function and a bidirectional LSTM followed by a dense layer. Each convolutional layer has 64 filters of size 8. This reference model generalized well for 135 GEOs.

The *cluster* models consist of a stacked version of the architecture of the reference model, with two convolutional layers and two stacked bidirectional LSTMs. Each convolutional layer also has 64 filters of size 8 with ReLu as the activation function and add a final dense layer. After the double $1D$ convolution spans the characterization of the input sequence, the first LSTM encodes such a characterization in states of 64 dimensions (bidirectional) and feeds into the second LSTM, whose units can now operate at a different time scale. This added complexity enabled the models to perform well in the GEOs where the reference model did not. After model selection, we obtained a bank of eight different cluster models.

4.2.2.2 Action branch

In this case we set a configuration with one LSTM followed by two dense layers to smooth the output. As in [27], a sigmoid activation function was used to fit the action layer's output in the range [0,1]. Considering that the increase of the stringency of an NPI should not decrease its effectiveness, g is constrained to satisfy the condition: if $\min(A - A') \geq 0 \longrightarrow g(A) \geq g(A')$. This constraint is enforced by using the absolute value of all trainable parameters of g after each parameter update. In order to keep the raw NPI constraints no convolution is considered here. The ValenciaIA4COVID predictor just takes into account the confinement NPIs, so each A_n^j is an 8-dimensional vector with the level of activation of the eight confinement NPIs (see Table 4.2).

4.2.2.3 Merge function.

Both the context and the action branches are fed with the data from the last 21 days that are combined into a final dense layer to get the predicted $\widehat{R}_n$. The outputs of each branch (h and g) are

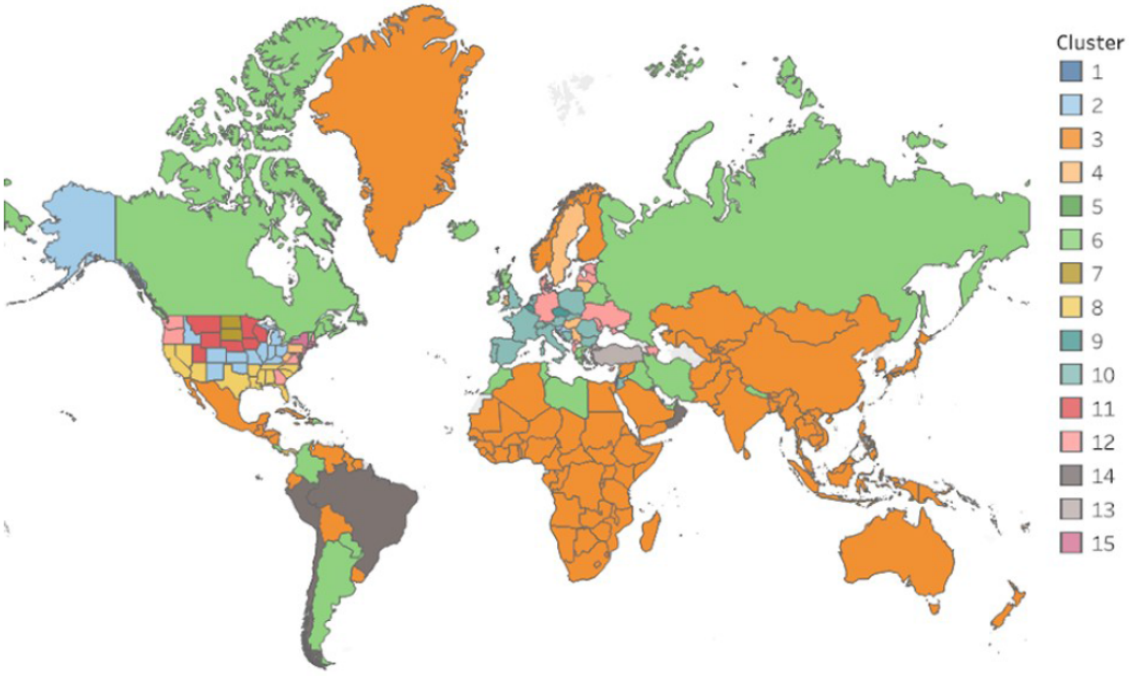


Fig. 4.6: Clusters obtained by using KMeans algorithm with $k = 15$, to the reported number of Covid-19 cases time series.

merged by the lambda function defined in 4.1. This means that the predicted $\widehat{R}_n$ provided by the *context* branch is modified by the output from the *action* branch. The stronger the NPIs (higher values), the larger the output from the action layer, thus reducing the context layer's output.

4.3 Results.

For comparison purposes, we benchmark our model with the LSTM architecture provided by the organizers of the XPRIZE Challenge trained with data from the 20 most affected countries. All models were trained with data between March 11th and October 31st for the long-term predictions, and between March 11th and November 30th for the short-term predictions. Below are the MAE per 100k and the MeanRank results when testing our model in the short-term (≈ 3 weeks ahead in December) and the long-term (7-8 weeks ahead since November 1st until December 21st before submission and December 27th after submission). In all experiments, the Valencia IA4COVID model outperformed the basic LSTM model in both metrics, as we show in Table 4.4.

These are the results we obtained with the bank of models we submitted to the Challenge, but soon after uploading the bank of models we realized that our predictor funcion was calling the wrong model in two countries (we had to adjust the models in a last call procedure, that is, few minutes before the deadline closed). Belize and Tunisia were not assigned to the right models, which in fact were present in the bank of models. These countries being miss-assigned caused our MAE per 100K grow in an undesired and unexpected way. We show the predictions for these two countries obtained with the wrong and the right models in Figure 4.9.

ValenciaIA4Covid team was awarded with the 1st place of the competition (`https://www.xprize.org/challenge/pandemicresponse/winners-results`) and $250.000 reward, being the first time an Spanish team won an XPrize foundation prize. We appeared in media and social networks, including the news at my alma mater `http://www.upv.es/noticias-upv/noticia-12746-ganadores-del-es.html`.

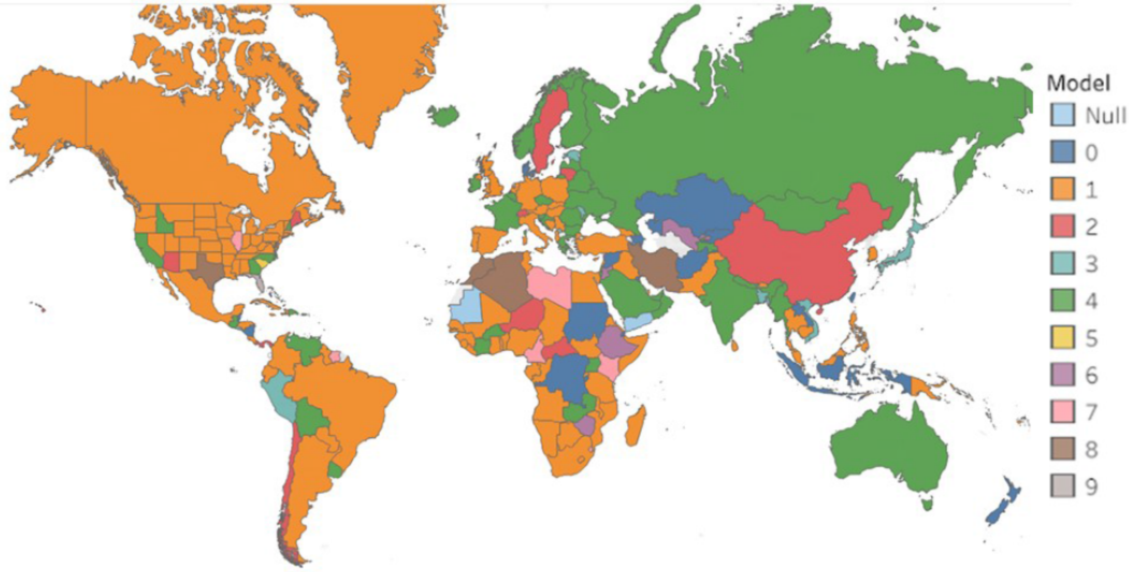


Fig. 4.7: Predictor model assigned to each GEO. Models 0 and 1 correspond to the reference model, 2 to 9 to the clustering models in our bank of models. Null corresponds to Mauritania and Yemen, GEOs in which we use an heuristic based on their historical Covid-19 cases because of the abundance of noise in their reported data.

Table 4.4: Short-term and long-term model evaluation.

		MAE per 100K		**MeanRank**	
	Dates	**LSTM**	**ValenciaIA4COVID**	**LSTM**	**ValenciaIA4COVID**
Short-term eval	**2020-12-01 - 2020-12-27**	241.646586	**163.031368**	1.557692	**1.442308**
	2020-12-01 - 2020-12-21	181.185431	**122.563651**	1.580508	**1.419492**
Long-term eval	**2020-11-01 - 2020-12-27**	922.339700	**935.791199**	1.587607	**1.412393**
	2020-11-01 - 2020-12-21	829.750857	**801.340524**	1.593220	**1.406780**

But most important, we provided to the local Government with predictions during the worst part of the pandemic, including evaluating different NPIs scenarios. Our models were able to predict the start (day) and the peak (number of cases) of the third wave of the Covid-19 virus infection. Our predicting models show to handle efficiently time series data, as it did with anomalous diffusing trajectories. Beyond the awards, the media exposure, the papers, beyond all that, we contributed to fight the pandemic.

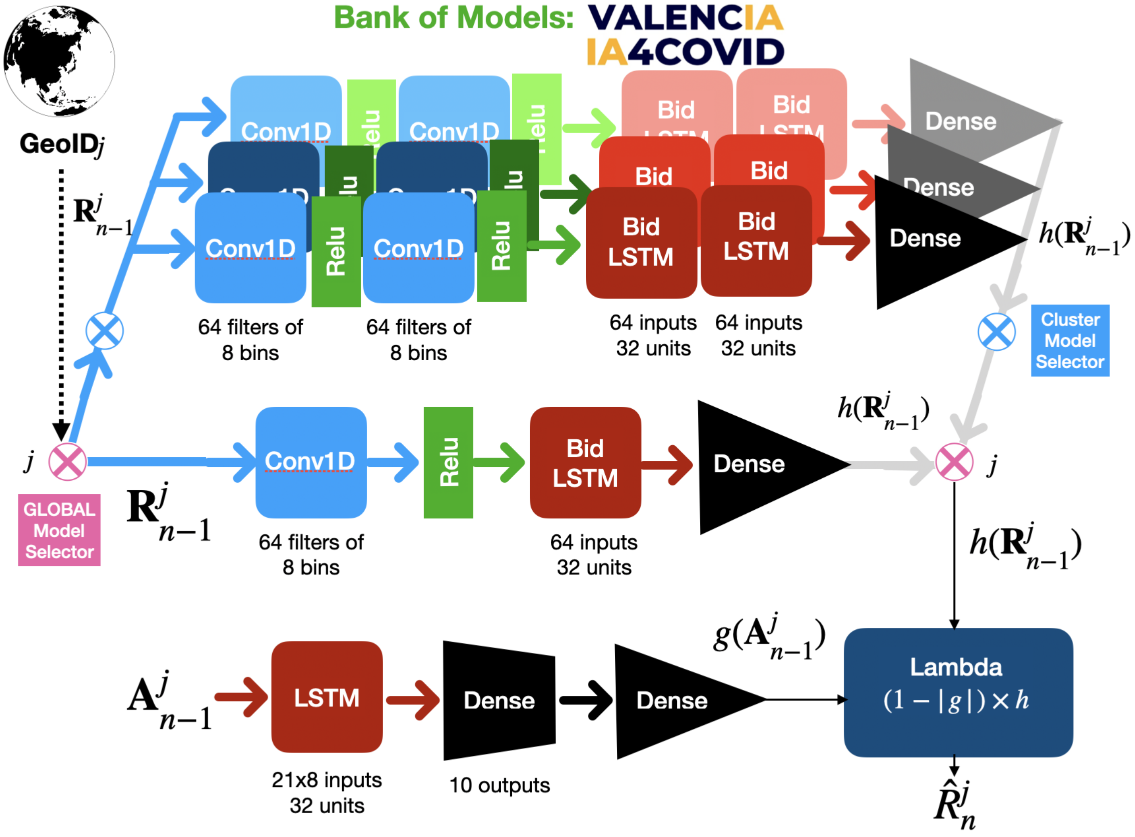


Fig. 4.8: ValenciaIA4Covid model, from our paper [20].

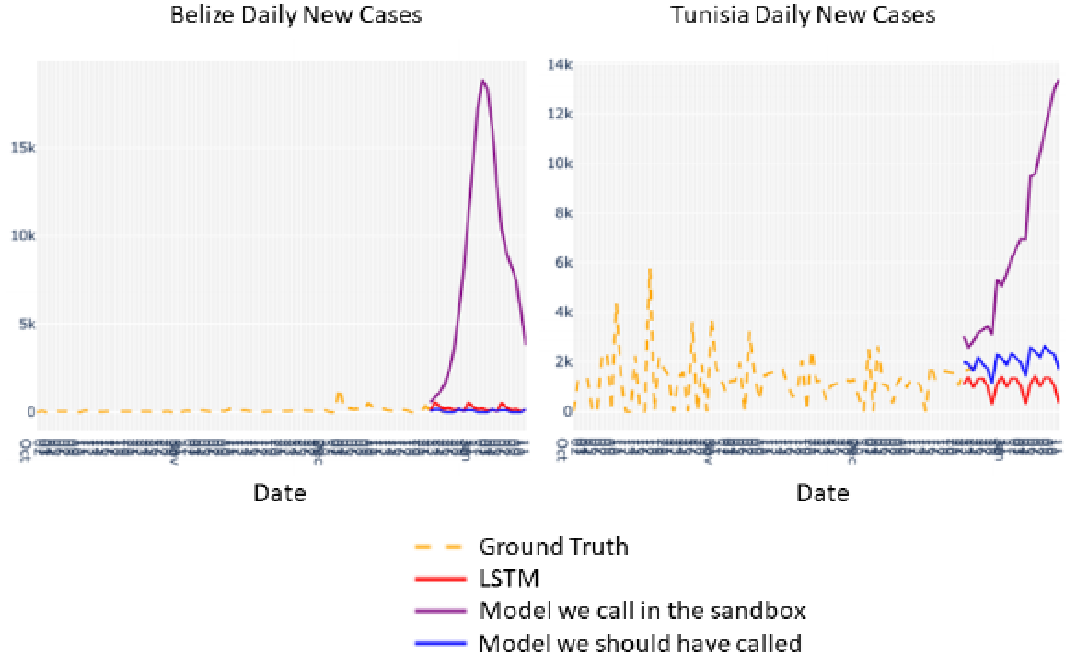


Fig. 4.9: Corrected and used model predictions for Belize and Tunisia.

5 Journal article (i)

It's the job that's never started
as takes longest to finish.

Sam Gamgee, The Lord of the Rings.

Efficient RNN methods for anomalously diffusing trajectories

Garibo-i-Orts, Ò[1], Baeza-Bosca, A.[1], García-March, M.A.[1], Conejero, J.A.[1]

[1] Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain.

Abstract. Anomalous diffusion occurs at very different scales in nature, from atomic systems to motions in cell organelles, biological tissues or ecology, and also in artificial materials, such as cement. Being able to accurately measure the anomalous exponent associated to a given particle trajectory, thus determining whether the particle subdiffuses, superdiffuses or performs normal diffusion, is of key importance to understand the diffusion process. Also it is often important to trustingly identify the model behind the trajectory, as it this gives a large amount of information on the system dynamics. Both aspects are particularly difficult when the input data are short and noisy trajectories. It is even more difficult if one cannot guarantee that the trajectories output in experiments are homogeneous, hindering the statistical methods based on ensembles of trajectories. We present a data-driven method able to infer the anomalous exponent and to identify the type of anomalous diffusion process behind single, noisy and short trajectories, with good accuracy. This model was used in our participation in the Anomalous Diffusion (AnDi) Challenge. A combination of convolutional and recurrent neural networks was used to achieve state-of-the-art results when compared to methods participating in the AnDi Challenge, ranking top 4 in both classification and diffusion exponent regression.

5.1 Introduction

Randomly moving particles, in some cases, diffuse anomalously in their surrounding medium. The concept of anomalous diffusion is defined in opposition to normal diffusion: since the movement is random, that is stochastic, the probability $P(\mathbf{x}, \mathbf{t})$ of finding a particle at time t and position $\mathbf{x} \in \mathbb{R}^{\mathbf{d}}$, $d = 1, 2, 3$ determines the dynamics. For normally diffusing particles, its width $\langle \mathbf{x^2} \rangle$, known as Mean Squared Displacement (MSD), grows linearly with time. This occurs e.g. in the traditional Brownian motion and is described by a partial differential diffusion equation (e.g. see a beautiful modern discussion on Fick, Einstein, and Smoluchowski Diffusion Equations in [29, 30]). If the MSD does not grow linearly with time, that is $\langle \mathbf{x^2} \rangle \propto \mathbf{t}^{\alpha}$, with $\alpha \neq 1$, then the particles following such movement are said to anomalously diffuse in their medium. The coefficient α is known as the anomalous diffusion coefficient.

A great variety of systems can show anomalous diffusive behavior. Furthermore, the theoretical models best explaining such systems are also extremely heterogeneous. For example, some of the models describe particle motion as a sequence of displacements of random lengths occurring at stochastic times, as in Brownian motion. Hence, both positions and times are stochastic variables whose behavior is determined by their corresponding Probability Distribution Functions (PDFs). This behavior occurs in a wide class of models termed as continuous-time random walks (CTRWs) [31]. A kind of CTRW showing anomalous diffusion is that in which the PDF describing the random time intervals between successive jumps is a power-law distribution $\psi(t) \approx t^{-\sigma}$, and displacements are sampled from a Gaussian PDF with variance D and zero mean. Another class of models is obtained when, on top of a power-law PDF for the waiting times, the PDF for displacements is not Gaussian [32]. These models are known as Lévy walks, and here the stochastic times are known as flight times. For example, in one dimension the displacements length is $|\Delta x| = |x_{i+1} - x_i|$, where x_i is the position at time t_i, are correlated with the flight times with the conditional probability $\Psi(\Delta x|t) = \frac{1}{2}\delta(|\Delta x| - vt)$ where v is the velocity. A model which results from the motion of a Brownian particle whose diffusion coefficient varies in time is the annealed transient time motion (ATTM) model [33]. Other models are obtained considering a variety of situations and geometries, like the bouncing of a particle in a set of regions with partially transmitting boundaries of stochastic heights [34], interactions between heterogeneous partners [35], the movement of a particle in an environment with critical behavior [36], etc. Another class of models can be defined from the Langevin equation: the stochastic differential equation governing the movement of a single particle with stochastic noise driving its movement (and modeling an environment interacting with the particle). Here, one may consider that the noise is non-white (termed as fractional Gaussian noise), with a normal distribution with zero mean but power-law correlations between the noise at different times. The resulting models are known as fractional Brownian motion (FBM) models [37, 38]. Yet another class of models is obtained when, in the Langevin equation, one considers time-dependent diffusivity, even with white Gaussian noise [39]. This is known as scaled Brownian motion (SBM). For the anomalous diffusing case, the diffusivity has power-law dependence with respect to t. See a review of anomalous diffusion models in [40].

The anomalous diffusing behavior is diverse and, indeed, it can be best explained with many different theoretical models. The behavior is very different attending at the anomalous diffusion coefficient, α. A limiting behavior occurs when α is close to 0, as then the width of the PDF describing the position and times of particles does not change in time, being regarded as a trapping situation. If α lies in the interval $0 < \alpha < 1$, the diffusion is called subdiffusive, while if $\alpha > 1$, it is called superdiffusive. The larger α we consider here is $\alpha = 2$, which is conventionally called ballistic motion and, of course, $\alpha = 1$ corresponds to normal diffusion. We do not consider here values of α larger than two, which are of course possible, and correspond to stochastic acceleration.

Then, a diffusing process of which one has access to the series of positions and times of a randomly moving particle, can be characterized by the anomalous diffusion coefficient and the model which betters explains its behavior. The tools which permit to do this characterization depend strongly on the availability of data. First, a possible situation is that we can guarantee the following two conditions in the experiment: i) a large quantity of long-enough trajectories can be obtained; ii) one can assure a homogeneity condition. This last condition means that all particles correspond to the same process over the whole experiment and can be assigned the same model and anomalous diffusion coefficient. In such a case, one can characterize the system performing an ensemble average between all trajectories [40, 41]. A second possible situation is that one can assure the following conditions: i) one can obtain a very long trajectory; ii) one can assure that the particle's behavior does not change during the whole experiment; iii) one can assure that the behavior is ergodic, that is, that, with sufficient time, one realization of the experiment explores all possible configurations of the system. In such a case, one can use time averages to extract information of the process. But a third possible scenario is that in which the experiment is such that it may happen one or more of the following: i) the trajectories one can access are short; ii) one cannot assure that all trajectories of different particles are homogeneous; iii) one can only access to one or a few trajectories. In such case, to assign a single short trajectory to a diffusion process characterized by a theoretical model and an α one has to find alternative tools to ensemble and time averages of high-quality data. One possible route is to use an approach based on a machine learning tool. In this paper, we present a high accuracy tool based on a particular kind of artificial recurrent neural network that has shown its utility and good performance when dealing with time series: the Long short-term memory (LSTM) architecture [5, 6].

Large theoretical efforts have built a battery of statistical techniques to find out the anomalous exponent given these difficulties. A non-comprehensive list includes Ref. [42] where a method based on the mean maximal excursion method was proposed, Refs. [43, 44] where a Bayesian estimation was proposed and tested for FBM processes, Ref. [45] where a method based in a fractionally integrated moving average was introduced, Ref. [46] where the large deviation theory of time-averaged statistics [47] was used to find some stochastic properties of measured trajectories, Ref. [48] where a statistical inference approach designed to find interactions between moving particles was used in experimental data, and Ref. [49, 50] where a method based in the information contained in the power spectral density of a single trajectory was proposed (for reviews and other methods see also [51–53]). Also, a Bayesian approach to test among different types of motion, which includes free motion (normal diffusion) and subdiffusion, was proposed in [54]. This paper considers other two types of motion: (i) *confined diffusion*, where particles cannot exit some structure, say a sphere in three dimensions [55]; and (ii) *directed motion*, where there is some flow in the ensemble of Brownian particles, for example, due to Brownian motors [56], which results in a ballistic MSD. On the other hand, statistical methods have been used to distinguish among models. For example in [57, 58] methods were introduced to distinguish among FBM and CTRW; in [59] an algorithm to identify and characterize FBM was introduced; Ref. [60] presents a Bayesian method to distinguish among Brownian motion, SBM and FBM (see also [61]); finally in Refs. [62, 63] it is discussed a method to distinguish among different physical origins for subdiffusion, which in turn point out to the different possible theoretical models.

Very recently, there has been a sudden growth of proposals that face this same problem with machine learning tools. A random forest classification algorithm was used to distinguish among directed motion, normal and anomalous diffusion, was introduced in [64] and was extended to include confined motion in [65]. A random forest was also used to classify trajectories as CTRW, ATTM, FBM, and LWs in [66] and also to assign an α single trajectories (see also [67, 68] where random forest and gradient boosting was used to classify among normal, super- and subdiffusion). In [69] a recurrent neural network was used to extract the exponent from a single short trajectory, even when the trajectory is sampled at irregular times. Also, in [14] a recurrent neural network is used to classify between the five models described above (CTRW, FBM, ATTM, LW, SBM) and obtain the anomalous exponent. In [70] a recurrent neural network was used to estimate the Hurst exponent of an FBM. A set of convolutional neural networks used to classify among Brownian motion, FBM, and CTRW, with simultaneous estimation of Hurst exponent H (which is related to the anomalous exponent α as $H = \alpha/2$) for FBM and the diffusion coefficient for Brownian motion, was presented in [71]. A convolution neural network was also used in [72] to classify trajectories as normal diffusion, anomalous diffusion, directed motion, or confined motion, and compared with random forest and gradient boosting. A combination of classical statistics analysis with supervised deep learning (a deep feed-forward neural network to cluster parameters extracted from the statistical features of individual trajectories) was used to classify among FBM, ATTM, CTRW, SBM, and LWs, and to infer α was introduced in [16]. A recent review on machine learning in the nearby field of active matter can be consulted in [73].

This research effort was the reason why the Anomalous Diffusion (AnDi) Challenge was launched in March 2020 (http://www.andi-challenge.org) [10, 74]. Similarly, as References [14, 16], the research described in this paper was a response to this challenge. The tasks in this challenge were, for short, noisy trajectories either in one-, two- or three dimensions: i) to propose and test a method able to distinguish among FBM, CTRW, ATTM, SBM, and LWs; ii) to propose and test a method to get the anomalous exponent. Here we present the tool which performed among the best ones in all these tasks and was best in the first task in one dimension. After a brief discussion of the experimental context for anomalous diffusion in next subsection, we describe in Sec. 5.2 the details of the method. In Sec. 5.3 we present the results obtained for both tasks, that is, inference of anomalous diffusion and classification according to theoretical model. We offer our conclusions in Sec. 5.4.

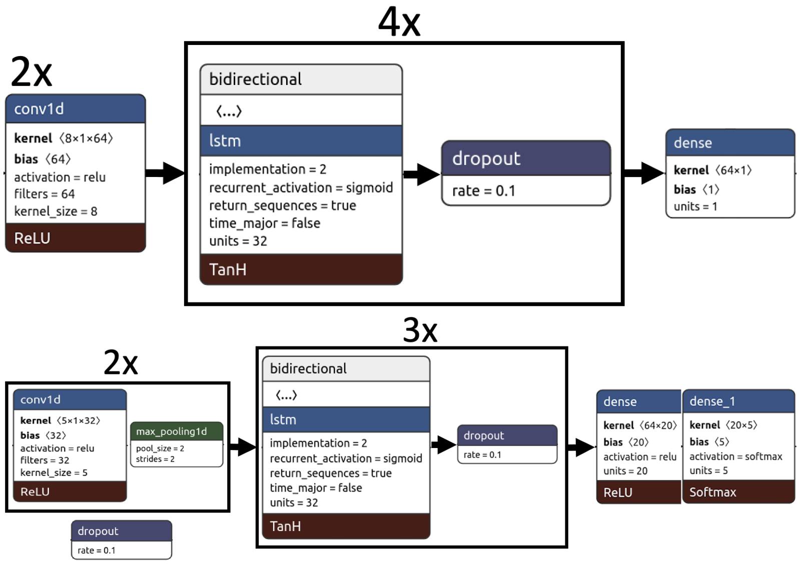


Fig. 5.1: *Schematic of the methods*. Top: Schematic of the method used for regression. Down: Schematic of the method used for classification.

5.1.1 Anomalous diffusion in experiments

Anomalous diffusion occurs in a plethora of experimental situations, ranging all scales [75, 76]. The smallest scale we are aware of occurs at the level of atoms, particularly in experiments with ultracold atoms [77–79] and also of quantized vortices in Bose-Einstein condensates [80]. Also, examples exist for ions in solutions [81]. Many experiments show anomalous diffusion in biological systems. For example, transient anomalous diffusion occurs for telomeres motion in the nucleus of cells [50, 82, 83]. Generally speaking, the recent developments in single-particle tracking techniques have boosted a revolution in cell biology [84], and several experiments have found anomalous diffusion, e.g. in the plasma membrane [85, 86] or in the cytoplasm [87–90]. Also, anomalous diffusion occurs in larger systems, like in worm-like micellar solutions [91], yeast cells [92], water in porous biological tissues [93–95], in cement-based materials [96], or ecology (see e.g. [97]).

The characterization of the kind of model that better explains the data obtained in an experiment and the associated anomalous exponent takes on key importance in many of these systems. For example, there has been large discussion on the underlying diffusion model and ergodicity which occurs in the experiments in [92, 98] (see [57, 99–101]). Also, since diffusion is the central transport mechanism in biological cells, if it is anomalous, it impacts how the system works. For example, it has been discussed that it may have an impact in chemical reactions [102]. Also, anomalous diffusion is compatible both with ergodic behaviors and non-ergodic behaviors, where a single realization does not explore all possible configurations (realizations) of the system. In the context of diffusion, one has weak ergodicity breaking if the averages taken over a single realization in infinite time do not equal ensemble averages over many realizations [40]. Also, for ultra-weak ergodicity breaking time and ensemble averages differ by a constant factor [103, 104]. Processes like CTRW, ATTM and SBM show weak ergodicity breaking [33, 86, 105], whereas Brownian motion and FBM are ergodic (this should be taken with care, as for example, it is shown in [106–108], the ergodicity of FBM requires a closer analysis). This is thus an important feature that may mark the ability to distinguish between models. Experimental signals are always noisy (e.g. in single-particle tracking, due to localization precision [109]). Often noise hides non-ergodic behavior [110] and hinders statistical analysis. Also, experimental trajectories are often short, depending of the kind of experiment [111]. Finally, one may not be able to assure that, in a biological system, that the measure trajectories are homogeneous. Therefore, a tool able to characterize diffusion from a single trajectory which is the output of an experiment, which is short and noisy, will find a great utility in a plethora of applications.

5.2 Description of the method

5.2.1 Generation of training and validation sets

The performance of a supervised machine-learning technique depends greatly on the quality of the data used to train. In this subsection, we discuss how we designed the training and validation sets, with the goal to have an homogeneous enough training set, that is, containing enough instances of all five models and a variety of anomalous coefficients, α, and trajectories lengths, yet being not too big, and hence computationally tractable.

To build the training and validation data set, we used the code provided in the AnDi Challenge [10, 74] that is freely available at `https://github.com/AnDiChallenge` on GitHub. We thus considered trajectories generated by the five classes (ATTM, CTRW, FBM, LW and SBM), with lengths ranging from 10 to 1,000. Considering steps of size 10, it makes 100 possible different trajectory lengths. Taking into account that in some models, a change in the regime is likely to show at some point in the trajectory, multiplying by a factor of 100 we can ensure at least one trajectory of each type in each part of the trajectory. Not all models can cover the whole range of diffusion coefficients. We considered values of α in the range from 0.02 to 1.95 with increments of 0.05, which means 19 different possible values for α. Note that the resulting set is imbalanced with respect to the models (e.g. CTRW is more represented than FBM in the subdiffusive regime, and LW is more represented than FBM in the superdiffusive regime). Putting all together, a data set of size 10^6 to ensure that, on average, at least one trajectory of each possible type will be present in the data set. We also consider trajectories with signal-to-noise ration (SNR) equal to 1 and 2, so this increases the recommended dataset size up to $2 \cdot 10^6$. We use then sizes equal or larger than this number. As it can be seen in [10, Fig.2] the performance of the models is not improved if we increase the SNR from 2 to 10.

In Table 5.1 we show the number of trajectories used for regression. In both tasks, we split available training data into training (90%) and validation (10%) independently at each epoch. The models are trained until no improvement was achieved after 10 consecutive epochs. We point out that when training and testing models were imbalanced. The reason is that when testing our models against the validation set provided by the organizers and some other validation datasets generated by us, we notice an improvement in the performance with the size of the training dataset. Moreover, it is worth mentioning that the validation dataset provided by the organizers contained only 10k trajectories, and in our case, our validation datasets were at least of size 200k trajectories.

5.2.2 Architecture of the method

The basic architecture used for both classification and regression consists of two convolutional layers used to extract spatial features from the trajectories. An initial convolutional layer is set with 32 filters with a kernel size of 5, making a sliding window of size 5 which slides through each trajectory extracting spatial features from them.

A second convolutional layer is used with the number of filters increased to 64 to extract higher-level features. Depending on the task, we reduce the dimensionality by applying a maxpool

Task	Trajectory length	1D	2D	3D
Regression	$L \in [10, 20[$	8	4	4
Regression	$L \in [20, 50[$	18	12	12
Regression	$L \in [20, 100[$	4	3	3
Regression	$L \in [100, 1000]$	2	2	2
Classification	$L \in [10, 1000]$	4	2	2

Table 5.1: *Number of trajectories per task, dimension, and length used as data set.* Number of trajectories scaled in millions, $\times 10^6$.

layer (in the classification task, not in the regression task) after each convolutional layer. The resulting encoded trajectories are fed in three stacked bidirectional LSTMs to learn the sequential information, with a drop-out layer of the 10% of the nodes to avoid incurring into overfitting. Different levels of dropout were considered while experimentation was in progress, ranging from 5% to 20%, being the selected 10% the one which outperformed the rest. Finally, we use several fully connected dense layers to predict the desired information (exponent regression or model classification).

5.2.2.1 Particularities of the method used for regression

We have used a trajectory length dependent approach by building models for different trajectory lengths. The following bins have been used based on trajectory length: $[10, 20]$, $]20, 30]$, $]30, 40]$, $]40, 50]$, $]50, 100]$, $]100, 200]$, $]200, 300]$, $]300, 400]$, $]400, 500]$, $]500, 600]$, $]600, 800]$ and $]800, 1000]$, what makes a total of 12 different models, all sharing the same architecture. We set two convolution layers followed by 4 stacked bidirectional LSTM layers. During experimentation having 4 stacked LSTM layers outperformed the 3 stacked LSTM option. We have two convolutional layers followed by 4 bidirectional LSTM blocks. After each block, a dropout layer is set. The output of the last one feeds a one node fully connected dense layer with linear activation function to get the estimated diffusion exponent (see Fig. 5.1, top panel).

5.2.2.2 Particularities of the method used for classification

Here we use a single model for all possible trajectories lengths and apply lead zero padding to each trajectory to make them of the same length (1,000). During the experimentation we found that applying dimensionality reduction layers (maxpool) at the output of each convolutional layer helped the LSTM layers to extract better sequential information to classify the trajectories, since doing so allowed to reduce the level of noise in the extracted features. The output of the last LSTM layer feeds a fully connected twenty-nodes dense layer with Relu activation function to capture non-linearity. This layer is followed by the final five nodes dense layer with softmax activation function to obtain five different probabilities for each trajectory to belong to one of the five possible models (see Fig. 5.1, bottom panel).

5.3 Results

5.3.1 Inference of the anomalous diffusion exponent

Here we present the results for one dimensional trajectories, while this tool was also used in two and three dimensions. The results in two and three dimensions are qualitatively similar, so for clarity and brevity we choose to discuss only the one dimensional case. The tool is available in the web site of the AnDi Challenge `http://andi-challenge.org/` and therefore results in all dimensions can be accessed there. Additionally, the code is available in `https://github.com/AnDiChallenge/AnDi2020_TeamM_UPV-MAT`.

To evaluate the accuracy of our results, we calculate the Mean Absolute Error (MAE) between predicted numerical α_{num} and the ground truth value α_{GT}. For N trajectories in the test set we compute the MAE as

$$\mathrm{MAE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{j}=1}^{\mathrm{N}} |\alpha_{\mathrm{j,num}} - \alpha_{\mathrm{j,GT}}|, \tag{5.1}$$

where the subindex j refers to the j-th trajectory. In the test set we included $N = 2000$ trajectories of increasing length $L \in \{20, 30, 40, 50, 100, 200, 300, 400, 500, 600, 800, 1000\}$, resembling the binning carried out in the training.

We used a pool of models: trajectories generated with a CTRW, FBM, LW, SBM, and ATTM. The trajectories produced with ATTM and CTRW are subdiffusive, i.e. $0 < \alpha \leq 1$ while the trajectories produced with LWs model are only superdiffusive, i.e. $1 \leq \alpha \leq 2$. Those produced with SBM and FBM cover the whole range of anomalous exponents, $0 < \alpha \leq 2$. Each trajectory is corrupted with some noise. To this end we consider the standard deviation of the displacements σ_{D} and add some Gaussian noise, with a standard deviation σ_{noise}, which is some portion of the σ_{D}. The SNR is thus $\mathrm{SNR} = \sigma_{\mathrm{D}}/\sigma_{\mathrm{noise}}$. We evaluate moderate and high noise, that is $\mathrm{SNR} = 2$ and $\mathrm{SNR} = 1$, which therefore means that σ_{noise} is half of σ_{D} or coincides with it. In Fig. 5.2 we plot the MAE as a function of length for different lengths of the trajectory and the two different noise levels.

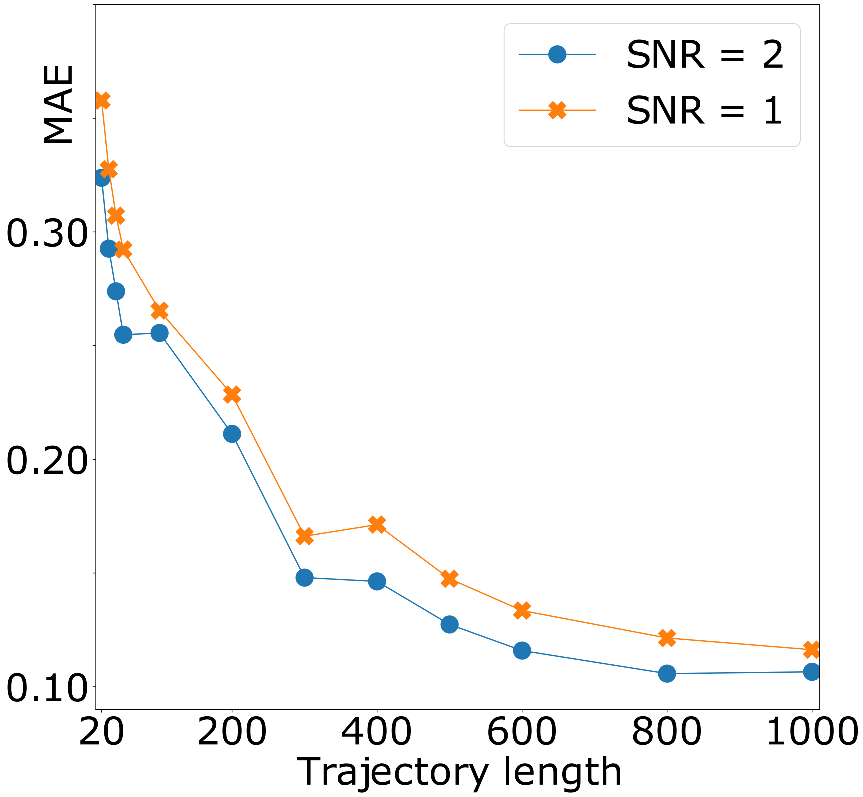


Fig. 5.2: *Inference of the anomalous exponent α as a function of length.* MAE as a function of length for $\mathrm{SNR} = 2$ and $\mathrm{SNR} = 1$ (that is $\sigma = 0.5$ and $\sigma = 1$).

As expected, the MAE gets better as the trajectories get larger. Mean absolute error improvement stabilizes around $L = 500$. Also as expected the results for $\mathrm{SNR} = 2\,(\sigma = 0.5)$ are better than for $\mathrm{SNR} = 1\,(\sigma = 1)$, for all lengths. For the length $L = 20$, with the current architecture, MAE between 0.3 and 0.45 are reached, which we set as a validity limit of the model. In Fig. 5.3 we plot the MAE as a function of length for the different models and those above two different levels of noise.

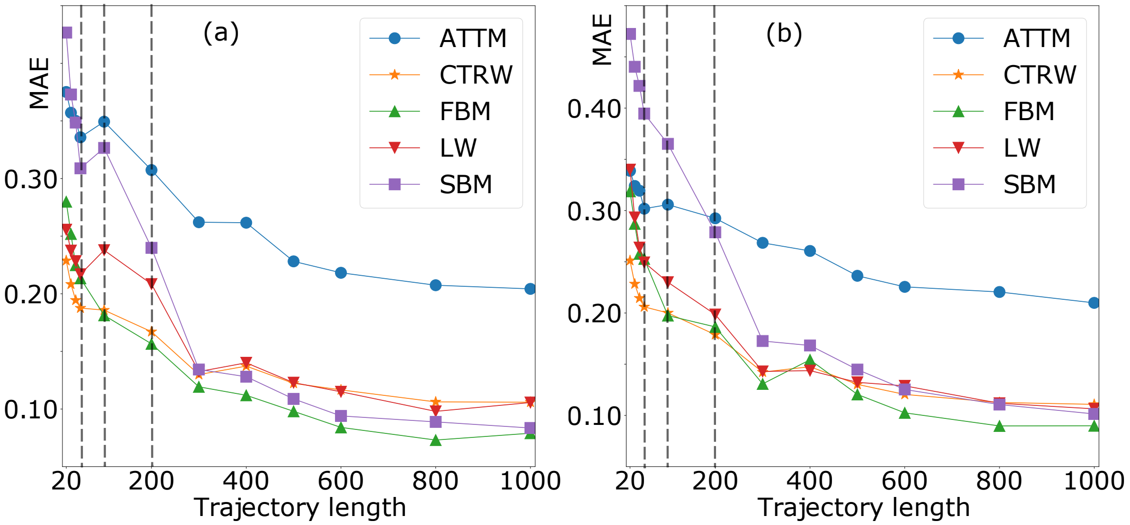


Fig. 5.3: *Inference of the anomalous exponent α as a function of length for different theoretical models.* MAE as a function of length for (a) SNR = 2 and (b) SNR = 1. Vertical dotted lines signal results at $L = 50, 100, 200$, to help discussion.

The results show an abrupt change in accuracy in the range $20 < L < 300$ for the SBM model for both levels of noise. For short trajectories ($L < 50$) and lower level of noise, $SNR = 2$, ATTM and SBM behave similarly but worse than CTRW, FBM, and LW. We recall that, by definition, both models display diffusive properties that vary with time and, therefore, they are the most challenging models to be determined. It seems that the deterministic time-dependent changes in the diffusivity of the SBM can be captured when trajectories are long enough $L > 300$. However, the ATTM is worse identified than the others, for $L > 200$, due to the random diffusion coefficient changes in time. This is probably because deterministic changes can be better identified and random changes as long as enough information is provided. Surprisingly, for larger noise ($SNR = 1$) and short trajectories ($L < 50$), ATTM have similar performance as the rest of models, while SBM keeps performing worse than any model. Nevertheless, ATTM reduces MAE as L is increased quite slowly ($L > 50$). Conversely, FBM, LW and CTRW quickly improve their MAE for short trajectories ($L < 50$), showing already reasonably good MAEs starting at lengths around $L = 100$. Finally, since ATTM, FBM, LW and CTRW show quite stable MAEs in the range $100 < L < 300$, we note that most of MAEs change in Fig. 5.2 in this range is due to the trajectories generated with the SBM model.

It is interesting to fix length, and have a closer look on how the model works for different values of the anomalous exponent. This is what we show in Fig. 5.4. Here, given a length L and a SNR value, we perform calculations for ground truth values of the anomalous exponent in the interval $\alpha_{\mathrm{GT}} \in [0.1, 1.9]$, in discrete increments of $\Delta\alpha = 0.1$. Here, we have also calculated the f_1-score, which is defined as $\mathrm{f}_1 = \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$, that is, twice the ratio of true positives (TP) over twice the true positive plus the number of false positives and false negatives of trajectories in the test set, N. We consider a TP when the predicted value of α, α_{num}, lies in the interval $[\alpha_{\mathrm{GT}} - \Delta\alpha/2, \alpha_{\mathrm{GT}} + \Delta\alpha/2]$.

With regards to the f_1-score, we observe that there is an abrupt change at normal diffusion, where $\alpha = 1$, which is more abrupt for long trajectories. We remark that not all models are used at all values of α (as we commented, ATTM and CTRW are restricted to $\alpha \leq 1$ and LW to $\alpha \geq 1$).

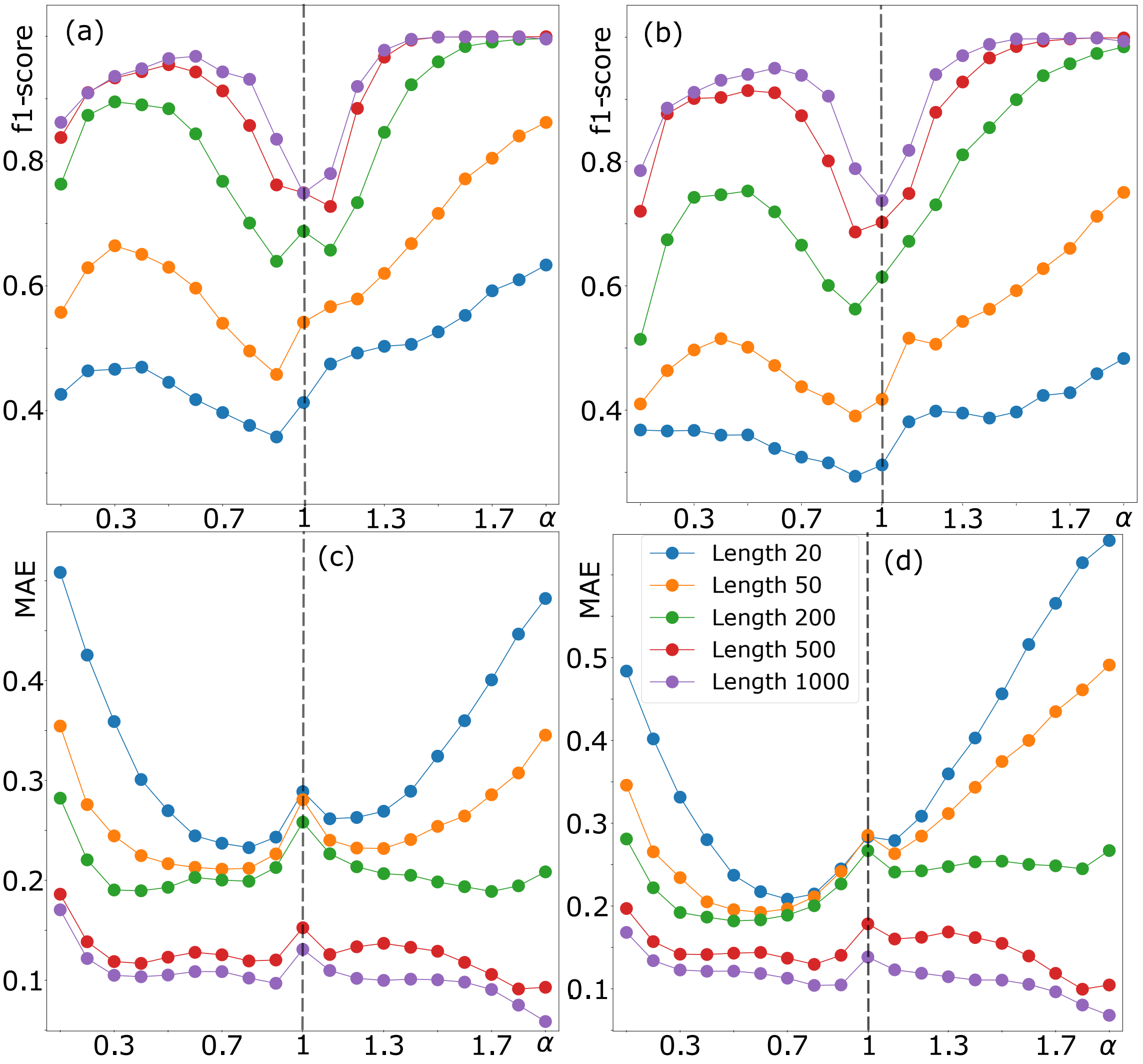


Fig. 5.4: *Inference of the anomalous exponent α for different values of the ground truth.* f_1-score and MAE for SNR $= 2$ (a) and (c) and SNR $= 1$ (b) and (d), for $L = 20, 50, 200, 500$ and 1000 (see legend) and trajectories in one dimension. Dotted vertical line signals normal diffusion.

We note then that at $\alpha = 1$ calculations are made with all five models. Also, the results are slightly better for $\alpha > 1$, which is also due to the existence of three models in this range, instead of the four models in the range $\alpha < 1$. Also, notice that below $\alpha = 1$ we consider the ATTM model, which as shown in Fig. 5.3 has a lower accuracy. We also appreciate, that the longer the trajectory is, the higher the f1-score is. However, from $L = 500$ up to $L = 1000$ it seems that the gain is very small when increasing the trajectory length. Respect to the MAE, the results are more or less stable around $\alpha = 1$ and get worse for shorter trajectories. Nevertheless, it shows pronouncedly that for short trajectories and closer to $\alpha = 0$ and $\alpha = 2$ the error increases. This also occurred in the figures for f_1-score, but only for $\alpha < 1$. For longer trajectories this effect is reduced and even inverted close to $\alpha = 2$.

We plot in Fig. 5.5 the distribution of α predicted as a function of the ground truth α, for the pool of models and two different lengths and noise levels. Since the dataset is generated with an equal number of trajectories of each model, and there are 4 models in the subdiffusive regime and 3 in the superdiffusive one, there are more trajectories in the left part of the pictures. The spread around the correct value in the diagonal is similar at both sides of $\alpha = 1$. Also, the distribution is

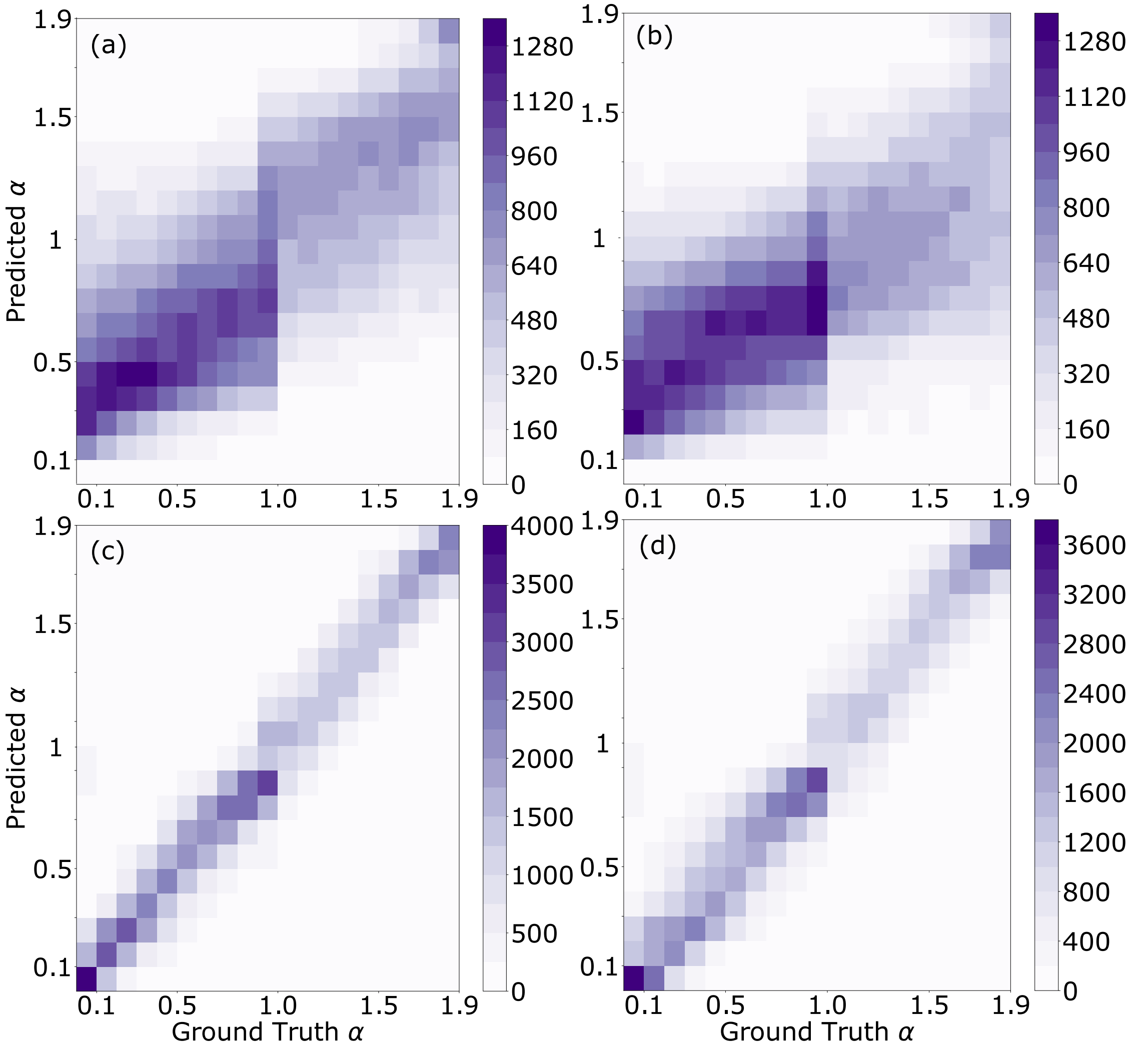


Fig. 5.5: *Distribution of the predicted anomalous diffusion coefficient as a function of the ground truth, for all models.* In (a) and (b) we plot the distributions for $L = 20$ and SNR = 2 and SNR = 1, respectively. In (c) and (d) we plot them for $L = 500$ and again SNR = 2 and SNR = 1, respectively.

wider for shorter trajectories than for longer trajectories (top panels versus bottom panels). Noise seems not to have a large impact in the results shown here. Finally, the decrease in the performance around $\alpha = 1$ agrees with the results in Figure 5.4.

To get further insight, we fix the length and SNR and we plot, for each model, the MAE and f$_1$-score for different values of the α in the interval $[\alpha_{\mathrm{GT}} - \Delta\alpha/2, \alpha_{\mathrm{GT}} + \Delta\alpha/2]$, for $L = 20$ and two different levels of noise (see Figs. 5.6 and 5.7). Notice that the range of α covered by each model is different.

In Fig. 5.6, we observe that the anomalous exponent inference of ATTM and SBM is poorer that in the rest of models. Similarly, the MAE gets worse as long as we approach to the limits of the exponent range: $\alpha = 0$ and 1 for ATTM and $\alpha = 0$ and 2 for SBM. On the contrary, the CTRW and LW, that they not cover the whole exponent range, respond better when approaching to the limits away different from $\alpha = 1$. Lastly, the anomalous exponent is easier to identify for the FBM in the subdiffussive regime.

In Fig. 5.7 we plot the same for $L = 500$. Here, the better predictions are obtained for the CTRW and LW models. The worst results are exhibited for the FBM and SBM around the normal

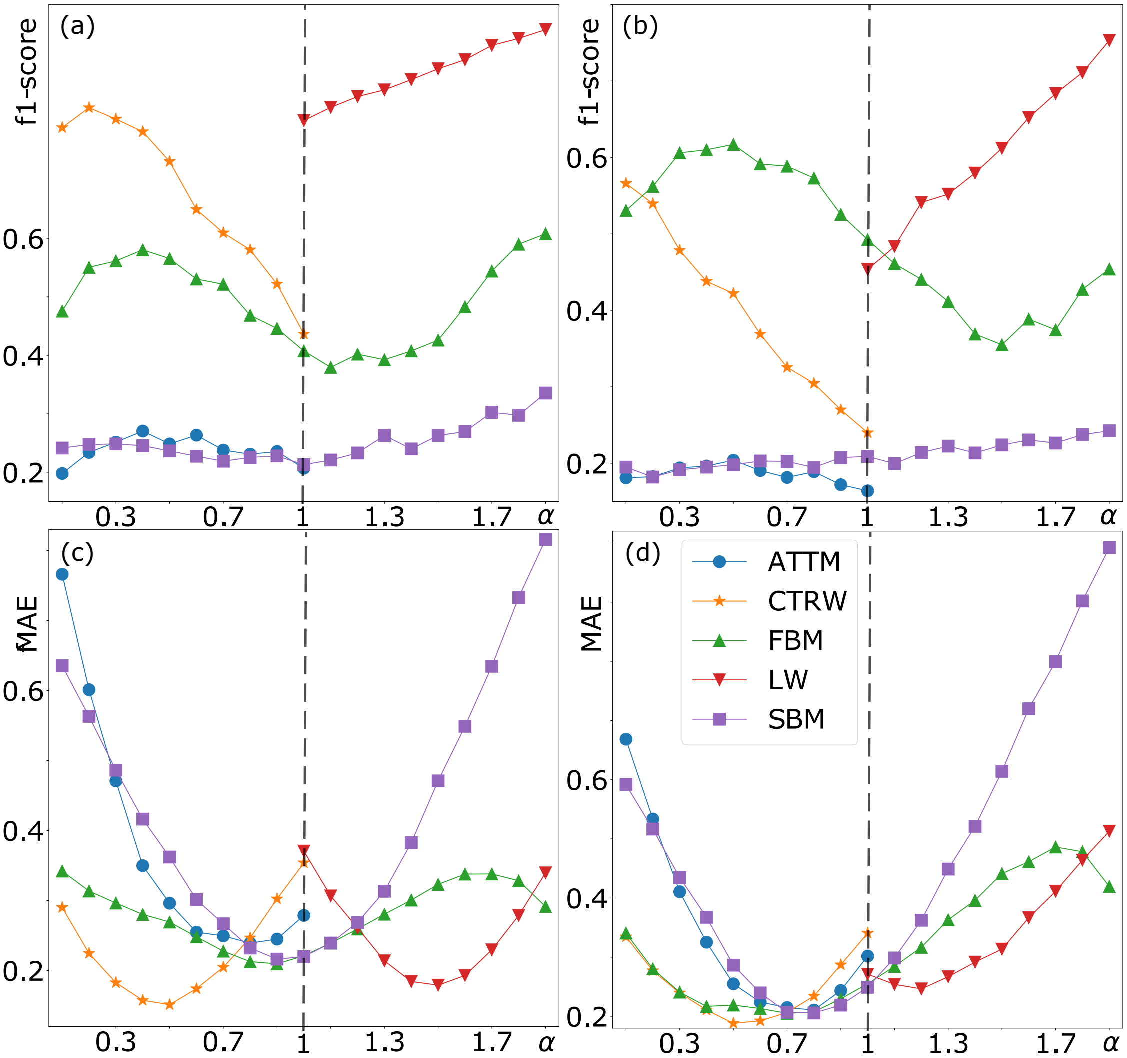


Fig. 5.6: *Inference of the anomalous exponent α for different values of the ground truth and for different theoretical models.* For $L = 20$, F_1-score and MAE for different values of ground truth anomalous exponent for SNR $= 2$ in (a) and (c); and the same for SNR $= 1$ (b) and (d).

diffusion regime. This is probably due to the fact that around $\alpha = 1$ we can find trajectories of all models, which confuses the predictions of the models that really exhibit trajectories with this diffusion exponent. We also see that the performance of the FBM decreases a lot around $\alpha = 0$ with high noise. Lastly, an increase of the length neither helps to improve the performance of the ATTM around $\alpha = 0$.

We finally plot in Fig. 5.8 the average of the α predicted as a function of the ground truth α for each model and two different lengths and noise levels, which disaggregates the results shown in Figure 5.7. Interestingly, in short trajectories, the predicted values of α in the subdiffusive regime tend to be higher than the ground truth. Conversely, in the superdiffusive regime, they tend to be slightly smaller than ground truth α. Nevertheless, we only find a clear bias for long trajectories in the ATTM model, which tends to predict smaller values than the real ones close to $\alpha = 0$, and in the *LW* around $\alpha = 1$.

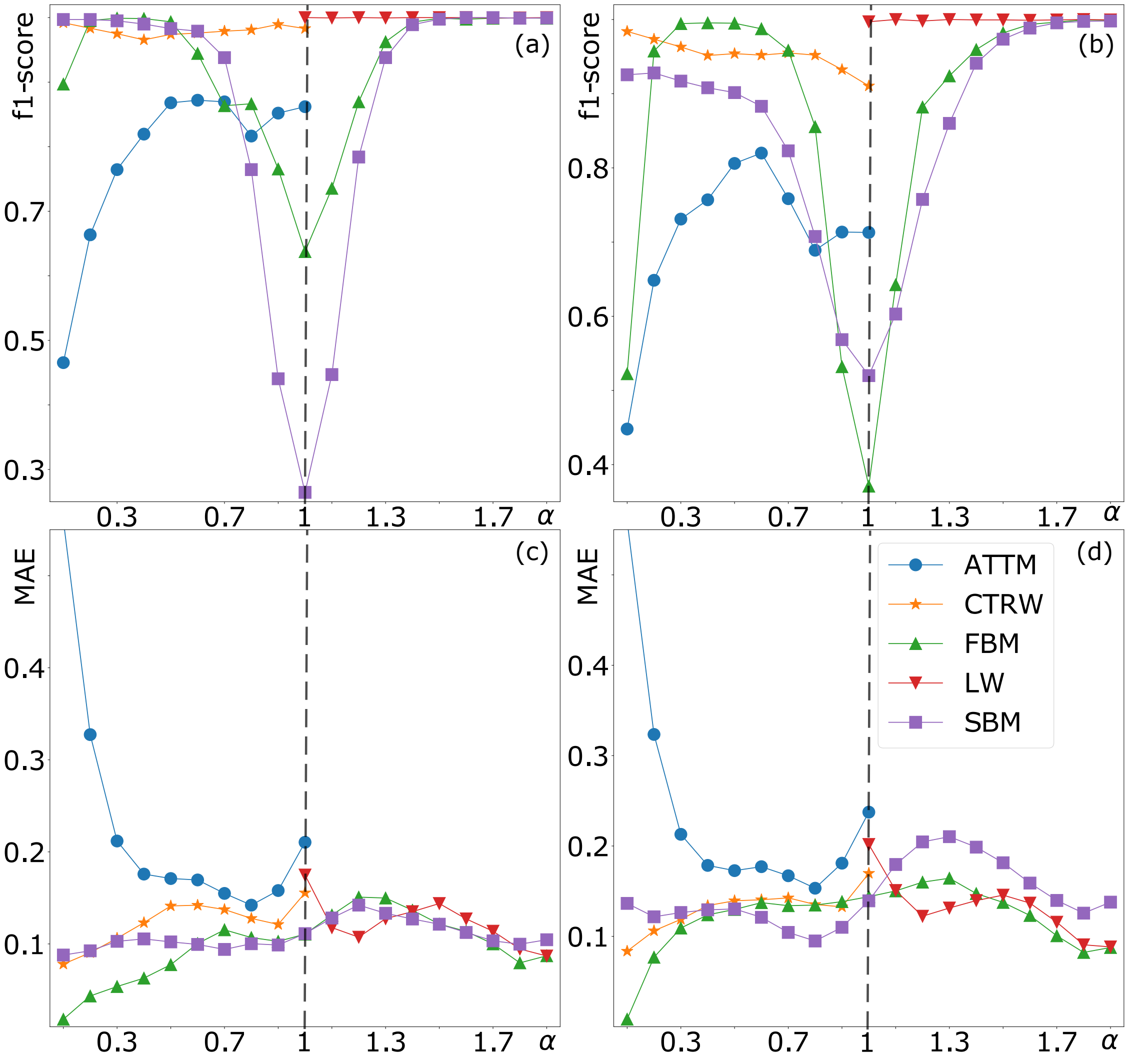


Fig. 5.7: *Inference of the anomalous exponent α for different values of the ground truth and for different theoretical models.* (a) F_1-score and (c) Mean absolute error for different values of ground truth anomalous exponent for $L = 500$ and SNR = 2; (b) and (d) same for SNR = 1.

5.3.2 Diffusion model classification

The second task is to predict the model which explains better the trajectory at hand. In Fig. 5.9 we plot the f_1 score as a function of trajectory length, showing the expected behavior (better results for longer trajectories and less noise). Similarly to the previous task there is a stabilization of the improvement of f_1 score around $L = 400$.

We also plot the f_1 score as a function of trajectory length for the different models and two noise levels in Fig. 5.10. Here, one can observe that the f_1 score is always larger for LWs and stabilizes in a large value for both levels of noise even at shorter lengths. Again, ATTM is the one that behaves the worst for all lengths. Finally, SBM also behaves worse than all other models except ATTM but, reminiscently of previous task, it improves faster than the rest of models for the larger level of noise.

We plot some exemplary confusion matrices in Fig. 5.11, for different lengths and noise levels. As shown before, the identification of the LW is very clear, even in short trajectories. In this case, the short trajectories of the rest of the models are often confused with the FBM. Again, the

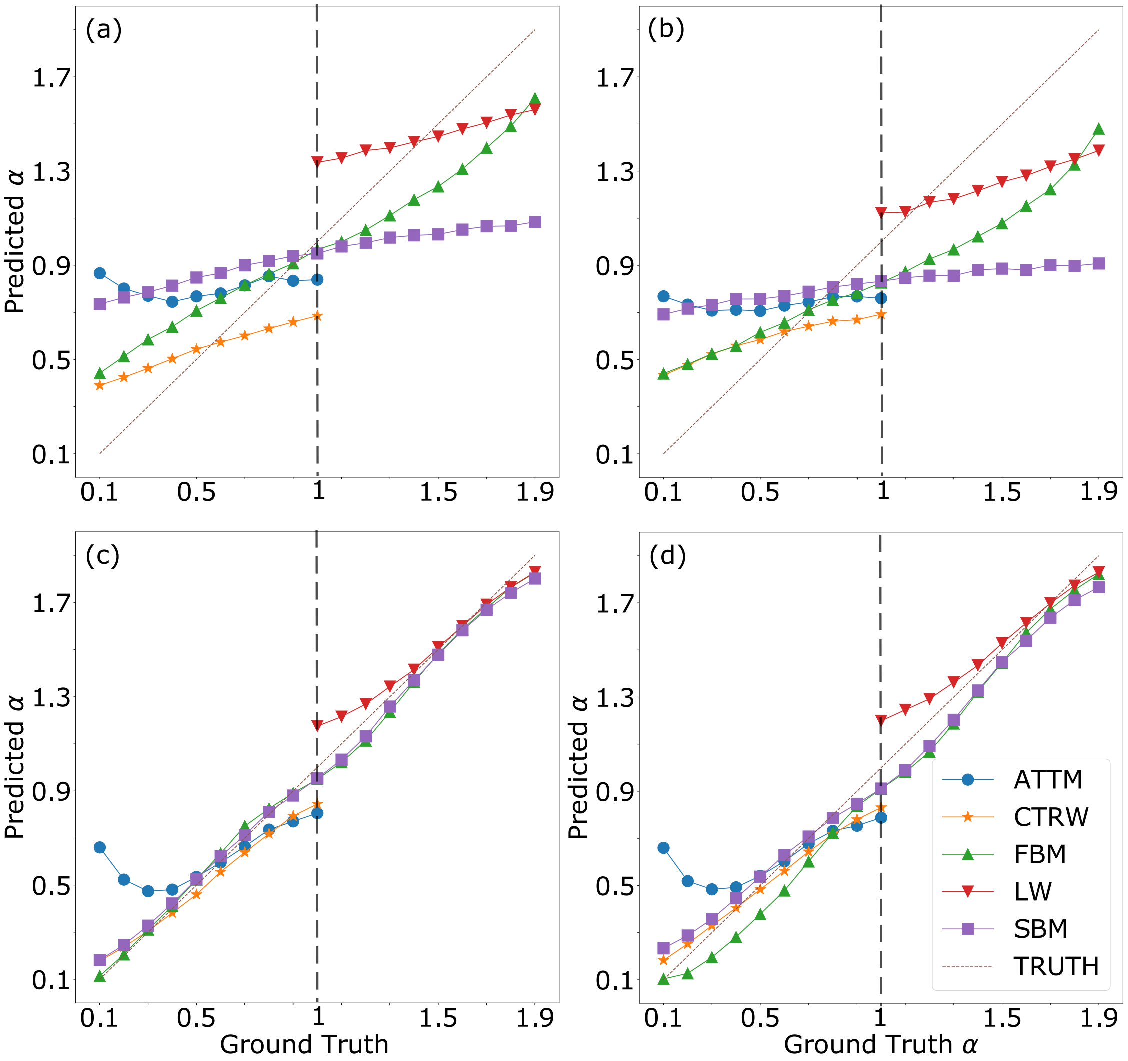


Fig. 5.8: *Average of the predicted amomalous diffusion coefficient as a function of the ground truth, for each models.* In (a) and (b) we plot the average for $L = 20$ and SNR = 2 and SNR = 1, respectively. In (c) and (d) we plot them for $L = 500$ and again SNR = 2 and SNR = 1, respectively.

CTRW and FBM are more accurately classified than the ATTM and SBM models. Finally, for long trajectories, we also observe that the ATTM is the worst classified method. We also note that the performance of the SBM was equally bad as the ATTM in short trajectories but increases quite a lot when the trajectories length increases.

Finally, we note that the results showed in Fig. 5.2 gave some hints on the model classifier. As commented, for moderate noise, $SNR = 2$, ATTM and SBM follow the same behavior (abrupt change below $L = 300$ and stabilization for $L > 300$). As we see now, the SBM is harder to identify with smaller noise than ATTM in short trajectories. However, the model classifies a little better ATTM. We can see that for short trajectories, with a larger SNR, the confusion between ATTM and SBM decreases.

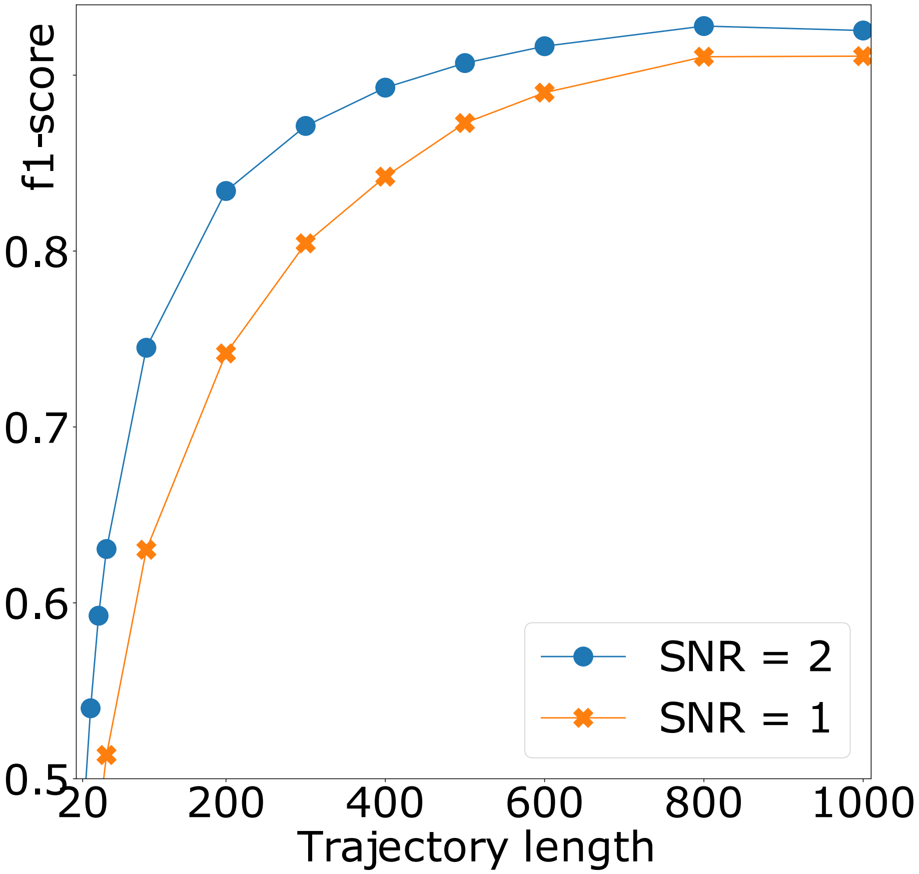


Fig. 5.9: *Diffusion model classification's f_1 score as a function of length* for two levels of noise in one dimension.

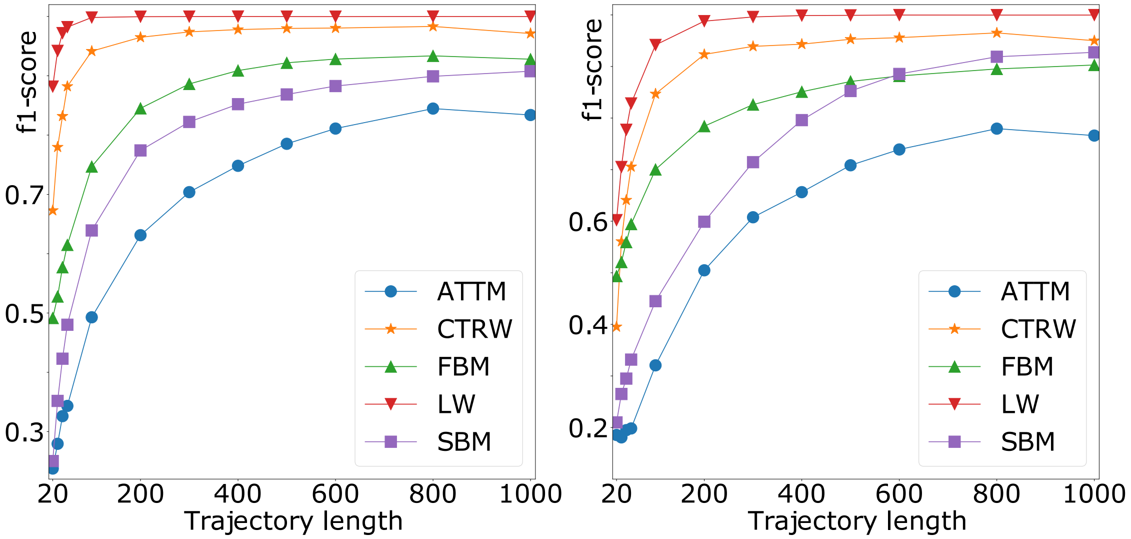


Fig. 5.10: *Diffusion model classification as a function of the trajectories' length for different models* f_1 score as a function of length for (a) SNR = 2 and (b) SNR = 1 for one dimension.

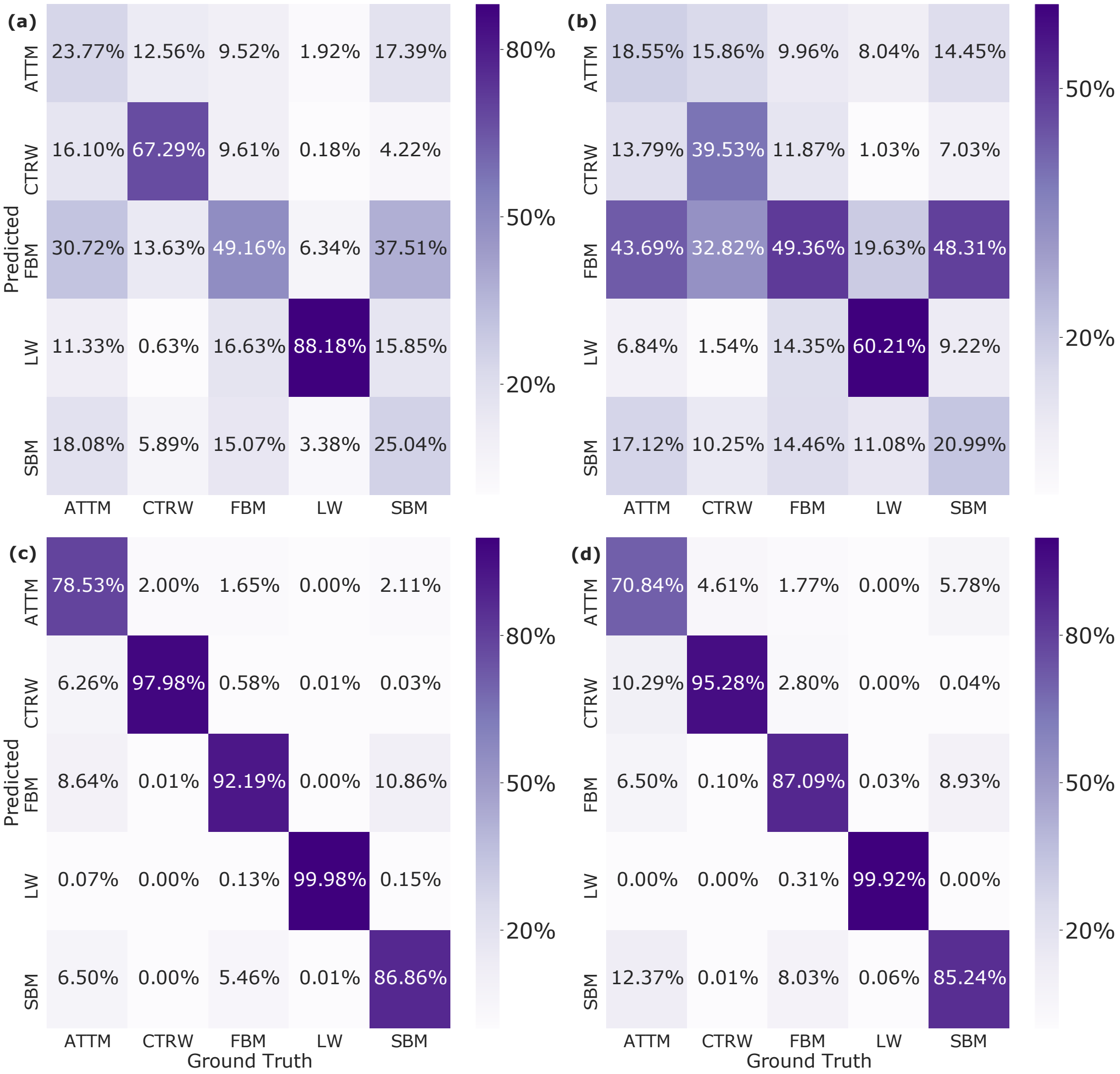


Fig. 5.11: *Confusion matrices*, calculated for $L = 20$ with SNR$= 2$ (a) and SNR$= 1$ (b), and for $L = 500$ with SNR$= 2$ (c) and SNR$= 1$ (d).

5.4 Conclusions

We presented the computational tool we used to participate in the AnDi challenge, which took place in 2020 (`http://www.andi-challenge.org`) [10, 74]. In the challenge the tools presented here ranked among the top 4 in all tasks, being the best in the regression task in one dimension. Besides, in the classification task, it ranked fourth in one dimension. With a similar model used with vectors containing all the trajectory coordinates, the model ranked third in two dimensions and second in three dimensions. The tool is a combination of convolutional and recurrent neural networks based on bidirectional LSTM blocks.

For task 1, i.e. inferring the anomalous exponent for a single trajectory, we obtain good MAEs below 0.2 for trajectories over length $L = 200$, even for the largest level of noise considered here. Also, for shorter trajectories we obtain reasonably good MAEs, finding a limit around $L = 50$. No further information is supposed to be at disposal when analyzing a single trajectory, if one wants to infer the associated anomalous coefficient. Anyhow, it is illustrative to study whether the MAE is different if we test the model only with trajectories generated with one model. We found that the worst performing for short trajectories are ATTM and SBM. Also, it is informative to see if the behavior error is larger if the analyzed trajectory has an α close to one or zero. We found that the error is much closer to normal diffusion, as expected, and in some cases closer to the limit of very trapped trajectories ($\alpha = 0$) and close to ballistic motion ($\alpha = 2$). Also, we showed that the dispersion on the predicted values of α is larger for shorter trajectories, as expected.

For task 2, i.e. classification of trajectories, the code should be able to assign a theoretical model to a given trajectory with large accuracy. Again, we found that we were able to obtain f_1-scores above 0.8 for long enough trajectories, with the accuracy dropping down for shorter trajectories and finding again a limit around $L = 50$. Again, no further information is supposed to be associated to the trajectory. But for academic information, we studied how f_1-score changes if we consider only trajectories of one of the models. We found again that the worst behaving models are ATTM and SBM. Finally, the confusion matrices show LW is easily identified and not confused with other models. This is to be expected as this model has peculiarities very different to other models, i.e. the correlation between step length and time waited. Also, all models are often confused with FBM. Finally, CTRW and FBM are more accurately classified than the ATTM and SBM models, and for long trajectories, the ATTM is the worst classified method.

In summary, the tool presented here offers good accuracies in both tasks. Dimensions two and three have not been discussed here, but the results are shown in [10]. The model shows, in higher dimensions, similar performances as the best performing methods in the challenge. As an outlook, we aim at using this tool in the task 3 of AnDi-Challenge, where one has a trajectory which changes behavior (anomalous coefficient or diffusion model) in an intermediate point, and the goal is to find accurately this point. First trials with our tool showed good results, and we will explore this in the future.

Acknowledgements

J.A.C. acknowledges support from ALBATROSS project (National Plan for Scientific and Technical Research and Innovation 2017-2020, No. PID2019-104978RB-I00). M.A.G.M. acknowledges funding from the Spanish Ministry of Education and Vocational Training (MEFP) through the Beatriz Galindo program 2018 (BEAGAL18/00203) and Spanish Ministry MINECO (FIDEUA PID2019-106901GBI00/10.13039/501100011033).

6 Journal article (ii)

A room without books is like
a body without a soul.

Marcus Tullius Cicero.

Gramian Angular Fields for leveragig pre-trained computer vision models with anomalous diffusion trajectories

Garibo-i-Orts, O.[1,3], Firbas, N.[4], Conejero, J.A.[2], Sebastia, L[3]

1 GRID - Grupo de Investigación en Ciencia de Datos, Valencian International University - VIU, Carrer Pintor Sorolla 21, 46002 València, Spain.

2 Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain.

3 VRAIN - Valencian Research Institute for Artificial Intelligence, Universitat Politècnica de València, Camí de Vera s/n, 46022 València, Spain.

4 DBS - Department of Biological Sciences, National University of Singapore
16 Science Drive 4, Singapore 117558, Singapore

Abstract. Anomalous diffusion is present at all scales, from atomic to large scales. Some exemplary systems are; ultra-cold atoms, telomeres in the nucleus of cells, moisture transport in cement-based materials, the free movement of arthropods, and the migration patterns of birds. The characterization of the diffusion gives key information about the dynamics of these systems, and provides an interdisciplinary framework with which to study movement. Thus, the problem of identifying underlying diffusive regimes and inferring the anomalous diffusion exponent α with high confidence is critical to physics, chemistry, biology, and ecology.

Classification and analysis of raw trajectories combining machine learning techniques with statistics extracted from them have widely been studied in the Anomalous Diffusion Challenge (Muñoz-Gil et al, 2021). Here we present a new data-driven method for working with diffusive trajectories. This method utilizes Gramian Angular Fields (GAF) to encode one-dimensional trajectories as images (Gramian Matrices), while preserving their spatio-temporal structure for input to computer-vision models. This allows us to leverage two well-established pre-trained computer-vision models, ResNet and MobileNet, to characterize the underlying diffusive regime, and infer the anomalous diffusion exponent α. Short raw trajectories, of lengths between 10 and 50, are commonly encountered in single-particle tracking experiments and are the most difficult to characterize. We show that by using GAF images we are able to outperform the current state-of-the-art, while increasing accessibility to machine learning methods in an applied setting.

6.1 Introduction

Computer vision models, by themselves, are unsuitable for sequence processing since the temporal structure of sequential data has few similarities with static images. An image of a dog is an image dog irrespective of its orientation. On the contrary, when dealing with sequences, like natural language, the order of words is highly relevant to the meaning of a sentence. Similarly, in a diffusive trajectory, the order in which features appear is highly relevant to the type of diffusion. Imagine that we have a sequence of confined diffusion. If we play the sequence backward, then it would appear as super diffusion, as the particle would increase its mobility with time.

Recurrent neural networks (RNNs) are widely used to combat the loss of temporal information when working with sequential data. However, RNNs have a key disadvantage since they must be trained sequentially. This greatly increases model training time, particularly when working with very large training data sets. Following Vaswani *et al.* 2017 [112], Transformers have largely taken the place of RNNs in natural language processing (NLP), with increased performance over RNNs and the ability to be trained in parallel [113]. However, there exists limited work in tokenizing and positionally encoding diffusive trajectories for use with Transformers, which prevents easy deployment of these models with one-dimensional data.

The difficulty and impracticality of designing, training, and tuning a custom machine learning (ML) model for a limited use scenario, where no pre-trained models are available, often discard the use of ML models in experimental settings. This has traditionally been true in diffusive studies, where the entry barrier to deploying ML models for the characterization of anomalous diffusion largely kept them out of the field. In fact, until the Anomalous Diffusion Challenge (AnDi Challenge[1]) 2020 [10], it was widely speculated that traditional statistical analysis could outperform ML-based methods. For instance, it took several weeks for our group to develop our ML model (ConvLSTM) based on convolutional neural networks (CNN) [1], and Long Short-Term Memory networks (LSTM) [5]. While our model produced excellent results, it is easy to see how such a time-consuming process could prevent the deployment of a similar solution.

The AnDi challenge showed that short trajectories were the most difficult to work with in both classification of the underlying diffusive model and inference of the anomalous diffusion exponent α. In this work, we will address this difficulty directly by working with short trajectories of lengths between 10 and 50. By working with these short trajectories, we will show that by using Gramian Angular Fields (GAFs) to leverage pre-trained computer vision models, natively available in Keras, we can outperform custom state-of-the-art models in a way that is accessible to those without an extensive machine learning background.

The paper is organized as follows: In Section 6.2 we review some preliminaries about anomalous diffusion, some examples in which it can be appreciated in experiments, and a revision of some existing machine learning models used for analysing the diffusive nature of the trajectories. In Section 6.3, we show how to convert a trajectory into different image representations through Gramian Angular Fields. Then, in Section 6.4 we describe the generation of the trajectories for training and validation of the models. Later, in Section 6.5, we compare the results of the new GAF fed computer vision models to our previous ConvLSTM model presented in the Andi Challenge [17], and we will benchmark the best of these new models using the AnDi Interactive tool [2]. Finally, we draw some conclusions in Section 6.6.

[1] http://www.andi-challenge.org

[2] `http://andi-challenge.org/interactive-tool/`

6.2 Anomalous Diffusion

6.2.1 Intro to Anomalous Diffusion

In 1827, Brown discovered that pollen grains placed in a fluid would move randomly and diffuse throughout the medium [114]. Due to the stochasticity of the movement, the probability $P(x,t)$ of finding a particle at time t and position $x \in \mathbb{R}^d,\ d = 1,2,3$ is used to determine the dynamics of the particle. A traditional metric, relying on the underlying stochastic nature of diffusion, is the *Mean Square Displacement* (MSD). The MSD of a system (an ensemble of particles) is defined as the average width or variance of its trajectories $\mathrm{MSD}(0,t) \equiv \langle x(t) - x(0)\rangle^2$ with respect to two points in time. The MSD is assumed to be taken with respect to a time t and the initial time $t = 0$, as such it is commonly abbreviated as $\langle x^2\rangle$. For a single particle, we approximate MSD with the time average of the MSD (taMSD)

$$\mathrm{taMSD}(\tau) = \lim_{T\to\infty} \int_0^{T-\tau} \mathrm{MSD}(t, t+\tau)dt, \tag{6.1}$$

where τ is some time interval. If MSD grows linearly with time, $\langle x^2\rangle \approx t$, then we say that the system diffuses normally. In contrast, anomalous diffusion happens when the MSD does not grow linearly with time, $\langle x^2\rangle \approx t^\alpha$ with $\alpha \neq 1$, where α is known as the anomalous diffusion exponent.

The diffusive behavior of a particle is known to vary greatly with α. As such, it has commonly been used to characterize anomalous diffusion. In general there are two kinds of anomalous diffusion; sub-diffusion, when $0 < \alpha < 1$, and super-diffusion, when $\alpha > 1$. At the lower end of sub-diffusion, α close to zero, we have particle trapping. When α is small, the width of the probability density function (PDF) $P(x,t)$ governing a particle's displacements becomes small. With the subsequent loss of variance between displacements, we have particle arrestation. At the upper end of the super-diffusive exponent range, we have what is referred to as ballistic motion, for $\alpha = 2$, and hyper-ballistic motion for $\alpha > 2$ [115]. Ballistic motion is characterized by unimpeded movement in a single line. In this work we will only consider the case $0 < \alpha < 2$.

The exponent α alone cannot be used to characterize a trajectory. While a change in α does indicate a different diffusive pattern, it is possible to have different underlying diffusive behaviors with the same anomalous diffusion exponent. For example, messenger RNA (mRNA) trajectories in a living *E. coli* cell can have very similar anomalous diffusion exponents while having distinct trajectories [40]. As such, there exists a need to be able to further describe a trajectory beyond its α.

The need to further describe the underlying movement of a subject undergoing diffusion inspired the classification task of the AnDi challenge. Classification of a trajectory based on similarity to a well-understood underlying diffusive regime can give more detailed information, such as its ergodicity, than knowing the MSD or anomalous diffusion exponent α alone. Ergodicity informs us about a possible difference between the local behavior and global behavior of a particle. In the event that a trajectory is non-ergodic, researchers may find it important to investigate the local behavior of the particle further to discover the source of the ergodicity breaking. In movement ecology, ergodicity breaking can result from variation among individuals, changes in behavior in the same individual, or the inherent heterogeneity of the landscape where resources are not evenly distributed; see for instance [116].

Following the precedent set by the AnDi Challenge, and in order to facilitate benchmarking, we will consider five underlying diffusive regimes, which we have summarized below. Further details regarding the computational implementation of these models can be found in [10, 117].

- *Continuous-Time Random Walk* (CTRW) happens when a particle motion can be described as a sequence of displacements sampled from a Gaussian distribution with zero mean. The waiting times between displacements are sampled from a power-law distribution $\psi(t) = t^{-\sigma}$ [31].

- *Lévy Walk* (LW) can be considered a special case of CTRW where dispersal lengths are correlated with waiting times. The probability density function (PDF) that describes the random time intervals between successive jumps is a power-law distribution $\psi(t) \approx t^{-\sigma-1}$ (as in CTRW), and the probability of a dispersal of length $\mathit{\Delta} x$ at time t is $\mathit{\Psi}(\mathit{\Delta} x, t) = \frac{1}{2}\delta(|\mathit{\Delta} x| - vt)\psi(t)$ [10, 32].
- *Annealed Transient Time Motion* (ATTM) occurs when a particle undergoes Brownian motion, with Diffusivity coefficient D_i for an interval t_i. The length of the interval t_i varies with D_i such that $P_{t_i}(t_i|D_i)$ has mean $E[t_i|D_i] = D^{-\gamma}$. Thus, for $\gamma > 0$, we would expect to see longer periods of low dispersal Brownian motion punctuated by shorter periods of high dispersal [33].
- *Fractional Brownian Motion* (FBM) is defined by the Langevin equation, which is the stochastic differential equation governing the movement of a single particle with stochastic noise driving its movement. If the noise is not white but follows a normal distribution with zero mean and power-law correlations between the noise at different times (fractional Gaussian noise) [37, 38].
- *Scaled Brownian Motion* (SBM) also derives from the Langevin equation, but in this case, diffusivity depends on time, even with white Gaussian noise [39].

6.2.2 Anomalous diffusion in experiments

Anomalous diffusion happens in a broad range of experimental situations at all scales [75, 76]. At the lowest scale, anomalous diffusion has been stated in experiments with ultra-cold atoms [77–79]. Experiments also show anomalous diffusion in biological systems such as the telomers' motion in the cell's nucleus, where transient anomalous diffusion happens [50, 82, 83]. A revolution in cell biology has been boosted by the developments achieved in single particle tracking techniques [84] with experiments finding anomalous diffusion in the cytoplasm [87–89] or in the plasma membrane [85, 86].

One can also find anomalous diffusion in bigger systems such as living yeast cells [92], worm-like micellar solutions [91], water in porous biological tissues [94, 95], and cement based materials [96]. Anomalous diffusion can even be observed in the migration patterns of storks between Africa and Europe. Tail winds on the storks return to Africa speeds up their journey giving a higher α for the trajectory from Europe to Africa than from Africa to Europe [118]. Thus, given an experiment, it is of paramount importance to characterize the model behind the data that best explains it and to infer the associated exponent α. For example, there is an ongoing discussion about the ergodicity and diffusion models at experiments in [57, 92, 98–101].

In the context of diffusion, a process is said to be ergodic if the behavior of a single trajectory can be said to summarize the whole system. More concretely a system is ergodic if taMSD(τ) = MSD$(t, t+\tau)$. In experiments, the possibility to perform time averages for $\lim_{T\to\infty}$ does not exit. Therefore the integral in Equation 6.1 converges to a fixed constant. In this case, because Equation 6.1 cannot converge to all MSD$(t, t+\tau)$ simultaneously, the process is said to be trivially non-ergodic [119]. Another special case of ergodicity breaking is where taMSD and MSD differ by a constant factor. This is referred to as ultra-weak ergodicity breaking [103, 104].

The five diffusive processes explained in Section 6.2.1, have different ergodic properties. We know that CTRW, ATTM, and SBM show weak ergodicity breaking [33, 86], whereas Brownian motion and FBM are ergodic (the ergodicity of FBM requires a closer analysis [106–108]). Thus, understanding the underlying diffusive model of a process is important as it can help direct further study. For instance, if a trajectory follows ATTM it will be locally Brownian, with sharp changes of the diffusivity coefficient D. In the context of research it may be of interest to study the cause of the shifts in D, as this may shed light on an important behavior.

With regard to signals resulting from Single Particle Tracking (SPT), it is important to remember that these experimental signals have inherent noise, and localization precision error [109]. This noise is problematic, as it has been known to hide non-ergodic behavior [110] and interfere with statistical analyses. More broadly, the difficulties resulting from the realities of experimentation

–short trajectories, noisy trajectories, and few replications– necessitate versatile tools, like those based on ML, to be able to characterize diffusion in experiments.

6.2.3 Machine Learning and Anomalous Diffusion

The difficulty of characterizing anomalous diffusion has given rise to diverse statistical methodologies to infer the anomalous exponent for a given trajectory. To name a few: Bayesian estimation for FBM processes [44, 120]; a statistical inference approach for finding interactions between moving particles [48]; a method based on fractionally integrated moving averages [45]; and a method based in the information contained in the power spectral density of a trajectory [49, 50]. Similarly, statistical methods have been used to discriminate between different diffusion models. For example, in [60] the authors present a Bayesian method to distinguish between Brownian motion, SBM, and FBM; and in [57, 58] we can find different methods to differentiate between FBM and CTRW.

The disadvantage of such statistical methods, both for the inference of α and determination of the underlying regime, is that they are specific to a particular type of diffusion. Machine learning-based inference and characterization methodology can be more flexible in that, since they can be applied to trajectories without prior knowledge of the underlying diffusion regime. However, such flexibility comes with the cost of lower interpretability and explainability (black box effect). Thus, when possible, we should strive to use well-established statistical methodology as we are able to make additional inferences based on these metrics.

The new wave of ML methods can be seen as a direct response to the realities of experimental design, current Single Particle Tracking (SPT) technology, and the inherent noise that exists in nature. Some of the first works in this line, include the use of random forests (RFs), which let some interpretability. In [64] RF were used to discriminate between direct motion, normal, and anomalous diffusion. Also, RFs were able to classify trajectories as CTRW, ATTM, FBM, or LW and to infer the exponent α [66], and RFs together with gradient boosting trees were used in [67, 68] to classify among normal, super and sub-diffusive trajectories. Complementary, in [72], convolutional neural networks were used to classify trajectories as normal diffusion, anomalous diffusion, directed motion, or confined motion and compared with RFs and gradient boosting trees.

Beyond explanatory machine learning approximations, we also find works in which a combination of classical statistics analysis and supervised deep learning (a deep feed-forward neural network to cluster parameters extracted from the statistical features of individual trajectories) are used to classify among the aforementioned five diffusive model, and to infer the anomalous exponent α [16]. Some other approaches to both problems lay in the use of deep learning methods, mainly based on LSTMs [14, 17, 69]. The role of the LSTM layers can be exchanged with Transformers, as it has been recently shown in [19] with very competitive results for short trajectories.

6.3 Gramian Angular Fields

First discovered by J.P. Gram, Gramian Angular Fields (GAFs) provide a methodology for converting a time-series, sequence, or vectors, to a matrix representation while retaining the existing spatial and temporal relations between the terms. GAF were first used, together with Markov Transition Fields, to feed computer vision models in Wang et al. 2015, where they were used for time series classification and imputation [121]. Since then, it has been extensively used to perform such diverse tasks. We only indicate a piece of them: regression of the day ahead solar radiation [122], prediction of the myocardial infarction risk from electroencephalogram signals [123], epilepsy detection [124], classification of human activity from sensor data [125], analysis of near infra-red spectroscopy signals [126, 127] and photoplethysmography signals [128]. However, to the best of our knowledge, this is the first use of GAF using computer vision models with anomalous diffusive trajectories.

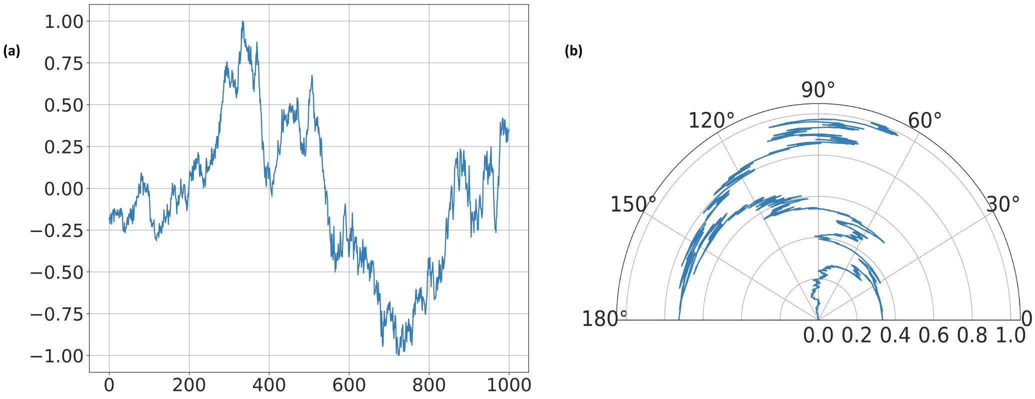


Fig. 6.1: A normalized super diffusive Lévy flight trajectory (a). (b) The polar encode of the trajectory in (a).

In order to encode a time series $\mathbf{x} = \{x_1, x_2, x_3, \ldots, x_N\}$ of length N, we first normalize $\mathbf{x}$, as $\tilde{\mathbf{x}}$ such that all its values belong to $[-1, 1]$, see (6.2).

$$\tilde{x}_i = \frac{(x_i - \max(\mathbf{x})) + (x_i - \min(\mathbf{x}))}{\max(\mathbf{x}) - \min(\mathbf{x})},\ 1 \leq i \leq N. \tag{6.2}$$

Once normalized, we convert each element in the sequence to polar coordinates. These are encoded by the angular cosine and are stored in $\boldsymbol{\phi}$. Likewise, the temporal positions of each value is stored in the radius $\boldsymbol{r}$ of the polar coordinates

$$\begin{cases} \phi_i = \arccos(\tilde{x}_i),\ \tilde{x}_i \in \tilde{\mathbf{x}} \\ r_i = \frac{i}{N},\ 1 \leq i \leq N, \end{cases} \tag{6.3}$$

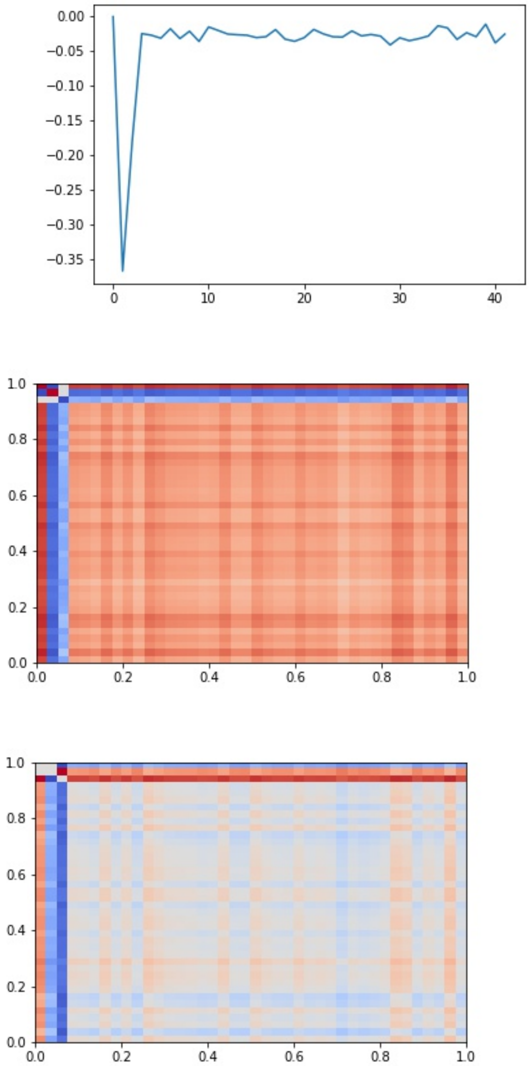
where N ensures that all $r_i \in [0, 1]$. As time increases, the values twist around the origin, as seen in Figure 6.1(b). This way of encoding a time series has two important properties: (1) it is bijective since $\cos(\phi_i))$ is monotonic when $\phi_i \in [0, \pi]$ and (2) the polar coordinates system preserves absolute temporal relations in contrast to Cartesian coordinates.

We can benefit from the angular perspective and consider the trigonometric sum/difference between each pair of points in the sequence in order to identify the temporal correlation within different time intervals. This yields two representations known as *Gramian Angular Summation Field* (GASF) (6.4) and *Gramian Angular Difference Field* (GADF) (6.5)

$$\text{GASF} = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \ldots & \cos(\phi_1 + \phi_N) \\ \cos(\phi_2 + \phi_1) & \ldots & \cos(\phi_2 + \phi_N) \\ \vdots & \ddots & \vdots \\ \cos(\phi_N + \phi_1) & \ldots & \cos(\phi_N + \phi_N) \end{bmatrix} \tag{6.4}$$

$$\text{GADF} = \begin{bmatrix} \sin(\phi_1 - \phi_1) & \ldots & \sin(\phi_1 - \phi_N) \\ \sin(\phi_2 - \phi_1) & \ldots & \sin(\phi_2 - \phi_N) \\ \vdots & \ddots & \vdots \\ \sin(\phi_N - \phi_1) & \ldots & \sin(\phi_N - \phi_N) \end{bmatrix}. \tag{6.5}$$

In Figure 6.2, we show a sequence in its raw format (A), and the GASF (B) and GADF (C) representations of that sequence. These GASF and GADF representations are then used to train two computer vision models: ResNet and MobileNet. To show how these representations maintain spatiotemporal relations, in Figure 6.3, we can see an example of a CTRW sequence (A), the same sequence backwards (B), and their GASF representations (C) and (D) respectively. Figure 6.3(D) is a 180 degree rotation of Figure 6.3(C). Additionally, both images are symmetric with respect to the main diagonal. This tells us that, for both GASF and GADF, time is encoded along the main diagonal of the GAF matrix and the x and y axes contain the spatial relations between terms of the sequence.



(a). Raw trajectory of length 40.

(b). GASF representation of raw trajectory

(c). GADF representation of raw trajectory

Fig. 6.2: We have a one-dimensional trajectory of length time 40 units (in a certain dimensionalized system scale) of a particle that is initially located in the origin (a). The time steps from 1 to 40 are indicated on the x-axis and the displacement of the points in the trajectory on the y-axis. In the GASF representation (b), the peak on the first time step in the trajectory is converted into 2 strong vertical and horizontal lines (index 1 and 2), as a result of the sum. In the GADF (c), we also have these stronger lines as a result of the difference, but in the rest of the picture, these differences are almost 0.

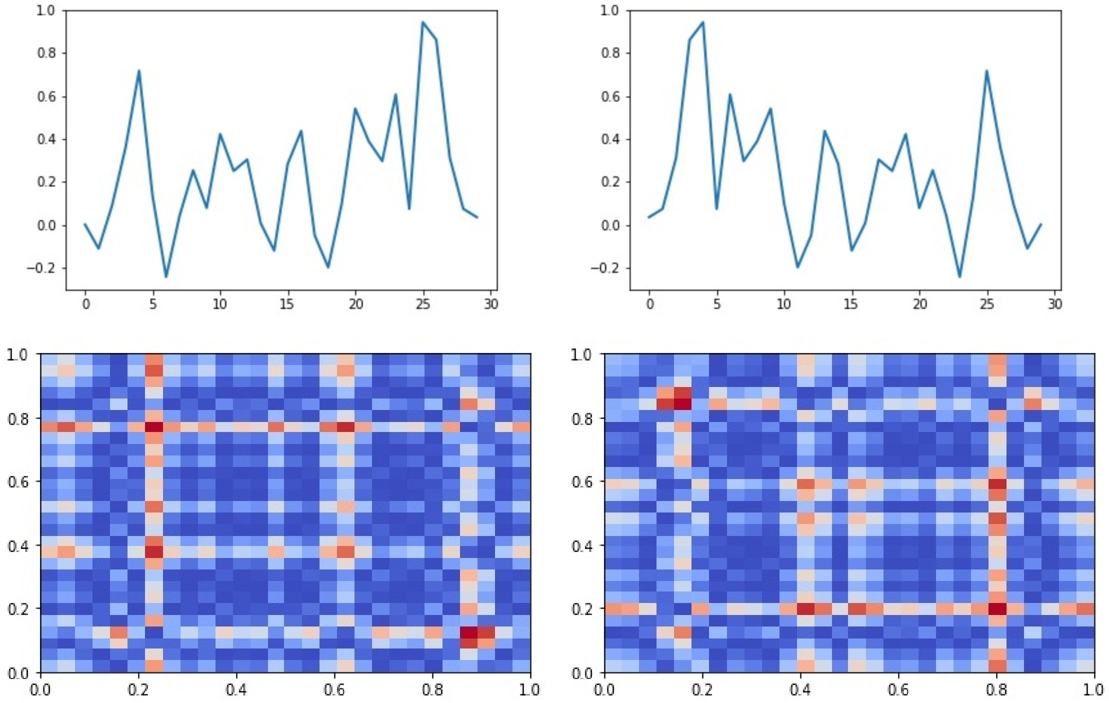


Fig. 6.3: In (upper left, A) we have a CTRW trajectory of length 30 time units and $\alpha = 0.2$. In (upper right, B), we have the same trajectory but reversed. In (lower left, C) and (lower right, D) we have the GASF image associated with the trajectory and its reversed copy. We see that these images are symmetric with respect to the main diagonal. Furthermore, we see that image (lower right, D) is a $\pi/2$ rotation of image (lower left, C), indicating that time is encoded along the same main diagonal.

6.4 Methodology

In this work, we use Gramian Angular Summation/Difference Fields (GASF/GADF) to represent trajectories as images in order to classify them according to their generating modeling to the five different classes (ATTM, CTRW, FBM, LW, and SBM) mentioned above (classification task) and for inferring the anomalous exponent α (regression task). In particular, we will focus on short trajectories of lengths 10 to 50, since these trajectories are the hardest ones to classify [10].

With the GASF/GADF images, we will train and validate two well-known models for dealing with the images ResNet [129] and MobileNet [130]. ResNet was created to address a loss in accuracy as convectional networks become deeper [129]. On the other hand, as the name implies, MobileNet is a small and efficient convolutional architecture that was designed to work well in mobile computer vision deployments. MobileNet is composed of depth wise separable convolutions and has two hyper-parameters, a width multiplier and resolution multiplier [130].

In order to benchmark our GASF/GADF fed Resnet and Mobilenet networks, we will primarily use the ConvLSTM method presented in [17], which placed in the top 2 at the AnDi Challenge 2020 both in underlying diffusion model classification and α regression tasks in one dimension [10]. In short, this method combines two convolutional layers, three bidirectional LSTM layers, and a last dense layer. It is described in Figure 6.4. Other models that present very good results can be found in [16], [14] [19].

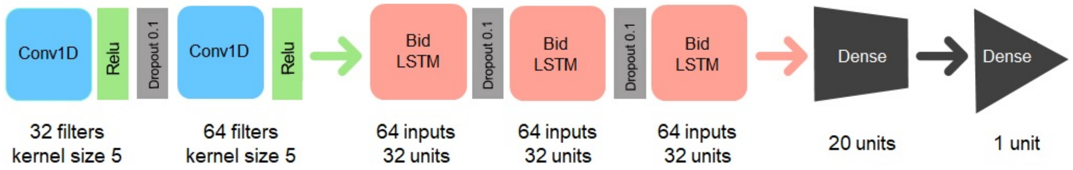


Fig. 6.4: ConvLSTM model architecture used for anomalous diffusion analysis in [10, 17].

6.4.1 Generation of training and validation datasets

The quantity and data training quality greatly determine the performance of supervised machine-learning techniques. We pretend to have enough samples of each of the five models and, simultaneously, cover the whole range of the anomalous exponent α. For generating the trajectories, we have used the code provided by the organizers of the AnDi Challenge [117] which is publicly available on GitHub at `https://github.com/AnDiChallenge`. The details on how these trajectories are generated can be found in [10, 117].

In order to train our GASF/GADF fed models we generated two different training data sets consisting of $4 \cdot 10^6$ trajectories each, of lengths ranging between 10 and 50. In order to work with images of the same size, all trajectories were padded with zeros at the end of the trajectory to ensure they were all of length 50. The first data set was used for classification purposes and considered the five aforementioned classes (ATTM, CTRW, FBM, LW, and SBM) as labels. The second one was built for the α exponent regression with $\alpha \in [0.05, 1.95]$ with increments of 0.05. In cases where the type of diffusion was only super or sub-diffusive a subset of the α interval was used. In both classification and regression, we have considered trajectories with Gaussian noise at $\mathrm{SNR} = 1$ and $\mathrm{SNR} = 2$. Gaussian noise has an standard deviation σ_{noise} which is some portion of the standard deviation of the trajectory displacements σ_D. We then define the signal to noise ratio (SNR) of a trajectory as $\mathrm{SNR} = \frac{\sigma_{\mathrm{D}}}{\sigma_{\mathrm{noise}}}$. This means that in our noisier trajectories ($\mathrm{SNR} = 1$) $\sigma_{\mathrm{noise}} = \sigma_D$, and in $\mathrm{SNR} = 2$ $\sigma_{\mathrm{noise}} = 1/2 * \sigma_D$.

In regression and classification, the training data sets were independently split into training (95%) and validation (5%) at each epoch. We trained the models until we got no improvement after ten consecutive epochs. Despite it breaks the 80/20 rule, the abundance of trajectories prevents us from over-fitting and permits us to deploy a more robust model. Finally, we have also generated two other data sets of 10^4 trajectories for testing the models and presenting the results.

6.5 Results

6.5.1 Diffusion model classification

The classification task consists in predicting which model best explains each trajectory best among five different classes (ATTM, CTRW, FBM, LW and SBM). It is worth mentioning that the trajectories following models ATTM and CTRW are always sub-diffusive, with α in the interval $]0, 1[$, while all the trajectories from model LW are super-diffusive, with α in $]1, 2[$. Finally, trajectories from FBM and SBM models take α in the full range $]0, 2[$. Gaussian noise was added to each trajectory to investigate the effect of noise in the classification. Throughout the discussion of the classification problem, we will just show the results for $\mathrm{SNR} = 1$. As expected, results will always be better for $\mathrm{SNR} = 2, (\sigma_{\mathrm{noise}} = 0.5)$ than for $\mathrm{SNR} = 1, (\sigma_{\mathrm{noise}} = 1)$, that is trajectories with lower noise are more accurately identified than noisier trajectories.

In order to study the performance of the different models, we consider the *F1-score* that is defined in (6.6), as the harmonic mean of precision and recall

$$\text{F1score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{6.6}$$

We remember that the *precision* measures the classifier's ability to correctly label positive samples; see (6.7)

$$\text{precision} = \frac{\sharp \text{ correctly predicted instances}}{\sharp \text{ predicted labels}} \tag{6.7}$$

and the *recall* measures the classifier's ability to find all the positive samples, see (6.8)

$$\text{recall} = \frac{\sharp \text{ correctly predicted instances}}{\sharp \text{ labels in the gold standard}} \tag{6.8}$$

It is worth mentioning that any of the combinations ResNet/MobileNet with GASF/GADF outperforms the benchmark ConvLSTM model in classification, for every trajectory length and noise level, as it can be seen in Figure 6.5 and in the Appendix. On average, GASF-GADF/ResNet models achieve the best results, so we will compare them against the ConvLSTM in detail.

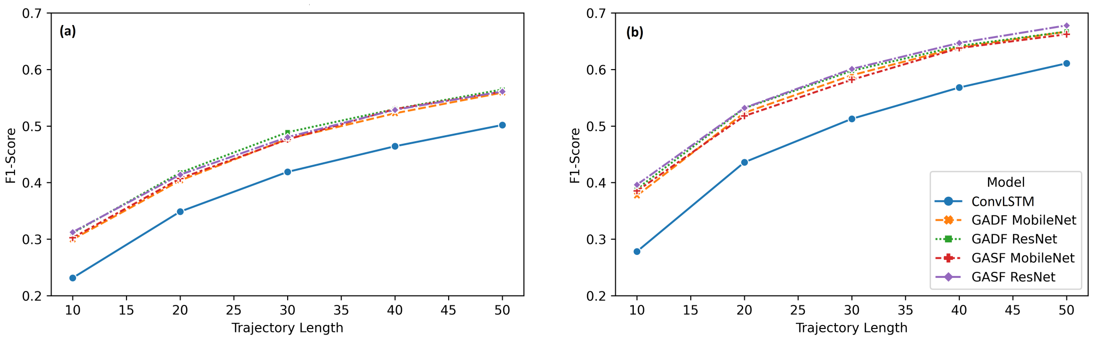


Fig. 6.5: F1-score is plotted against the trajectory length, with line color indicating performance for each of the ML models tested. In (a) we see the results for SNR = 1 and in (b) and the results for SNR = 2.

In Figure 6.5 we plot F1-score as a function of trajectory length and the ML model used. Our results from Figure 6.5 show that all of our GAF fed models are able to outperform our ConvLSTM in classification at any trajectory length. This is important as the ConvLSTM was the best overall model in one-dimensional classification at all trajectory lengths and at restricted lengths [10, 50] during the AnDi challenge (Table 6.1). When we focus on F1-score as a function of the anomalous diffusion exponent α and the trajectory length, as in Figure 6.6, we can see that F1-score improves with increased length. Though, if we focus on the interaction of length and α we can see that increasing length has a disproportionate effect on the F1-score for α close to 1.9 and for $\alpha \approx 0.3$. The source of this interaction is unclear. Though we do know that CTRW and LW classification performs best and the sub and super-diffusive extremes respectively, and FBM performs dramatically better at $\alpha \approx 0.4$ and $\alpha > 1.75$ (Figure 6.7). Thus, the interaction in question is likely due to the pooling of model performance across the five diffusive regimes.

Upon closer inspection of model performance by trajectory type (Figure 6.7, we see that the GASF and GADF ResNet models (in (a) and (b)) perform roughly the same. However, the difference in architecture between our GAF fed computer vision models and our ConvLSTM is immediately evident when we compare performance across the different trajectory types. Most notably, classification performance of SBM trajectories nearly doubles from the ConvLSTM to either ResNet Model. While performance across different trajectory types was found to deviate between the GAF fed ResNet models and the ConvLSTM the relative shape of the performance curves stayed the

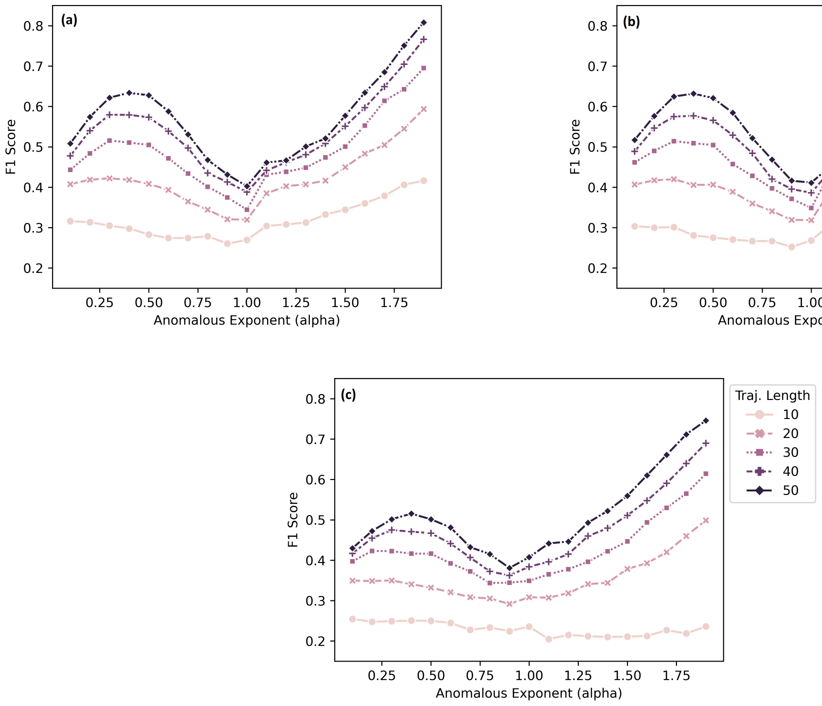


Fig. 6.6: F1-scores for the GASF/ResNet (a), GADF/ResNet (b), and ConvLSTM (c) models for different trajectory lengths and values of the anomalous exponent α for noisy trajectories with SNR = 1.

same. For instance, the curve of FBM F1-score α is roughly sinusoidal and the performance of CTRW and LW are approximately mirror images of each other. In particular, the association of F1score to α in CTRW and LW trajectories was also observed in the ConvTransformer [19]. Similar performance curve shapes across different ML architectures seems to indicate that certain values of α make some diffusive regimes appear as others. Though, it should be noted that all three of the mentioned architectures (ResNet, ConvLSTM, and ConvTransformer) have a convolutional component, so more would have to be done to determine that this is not the cause of the similarly shaped curves.

While overall performance was significantly improved with our GASF and GADF ResNet models over the ConvLSTM, the greatest improvements from ResNet came from $\alpha \in [0.25, 0.6]$ and from $\alpha >= 1.5$. With smaller improvements in the classification of trajectories with $\alpha \approx 1$ (Figure 6.6). It should be noted that there are two cases where GADF/GASF ResNet classification performance was worse than in the ConvLSTM. Most notably, the ResNet models have great difficulty with the identification of ATTM trajectories, with F1-scores dropping below 0.2, which is what one would expect from guessing. Then the sinusoidal shape of F1-score $\approx \alpha$ of the FBM trajectories is exacerbated in the ResNet models leading to a loss of performance around $\alpha \approx 1.25$, with performance gains elsewhere (Figure 6.7).

We have separately analyzed each diffusion type in terms of the trajectory length; see Figure 6.9. The GADF/ResNet model significantly outperforms the benchmark for any length and anomalous diffusion exponent α for all models. This can be clearly appreciated for LW and ATTM trajectories of lengths greater or equal to 30. The unique exceptions are ATTM and FBM trajectories of lengths higher or equal to 20. It seems that ResNet models try to concentrate on increasing the accuracy of CTRW, LW, and SBM diffusion types in comparison with ConvLSTM, which is more distributed within the five diffusion models.

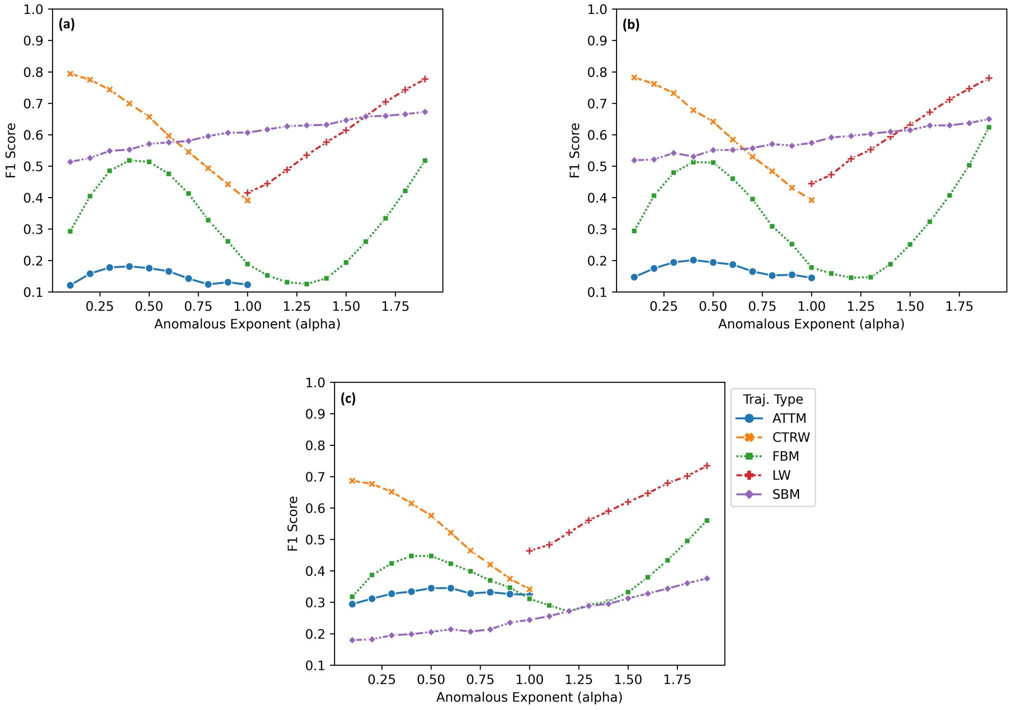


Fig. 6.7: F1-scores for the different models on noisy trajectories, SNR = 1, of all the five classes (ATTM, CTRW, FBM, LW, and SBM) with GASF/ResNet (a), GADF/ResNet (b), and ConvLSTM (c).

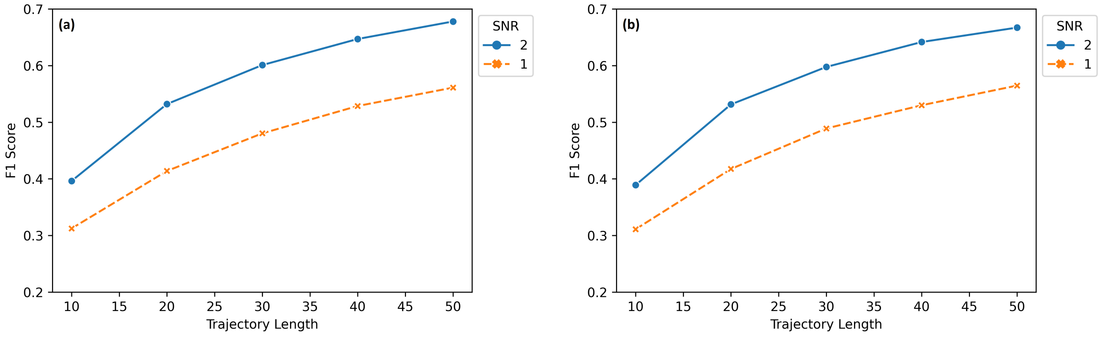


Fig. 6.8: F1-scores of GASF/ResNet (a) and GADF/ResNet (b) models for SNR = 1, 2.

Finally, we have bench-marked our model using the AnDi interactive tool for trajectories of length 10 to 50. The resulting confusion matrix can be seen in Figure 6.10 and it largely summarizes the results we discussed in this section, where the model is best at identifying CTRW and LW trajectories, with difficulty identifying ATTM trajectories. From the outputs of the AnDi interactive tool we can verify that our GADF and GASF ResNet models significantly outperform the previous top models (Table 6.1) in overall F1-score and AUC. We were able to achieve these performance increases, without sacrificing ease of implementation in order to mantain our goal of increasing accessibility to ML methods for characterization of anomalous diffusion. In fact, the GASF and GADF models were significantly easier to deploy than ConvLSTM previous state of the art, and as such should be very useful in an applied setting.

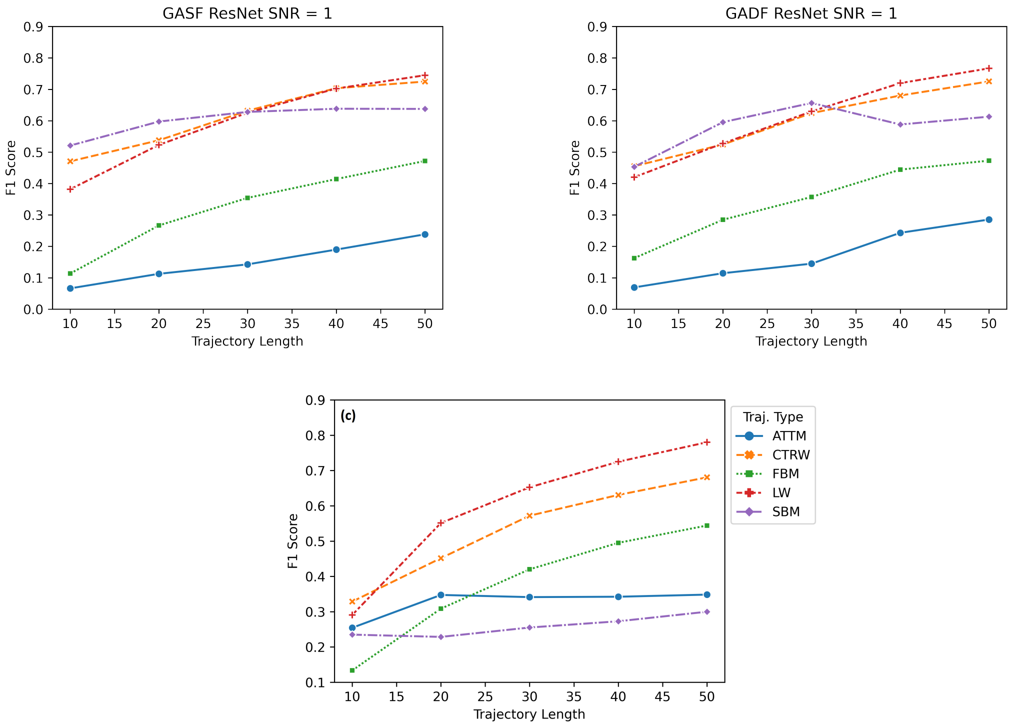
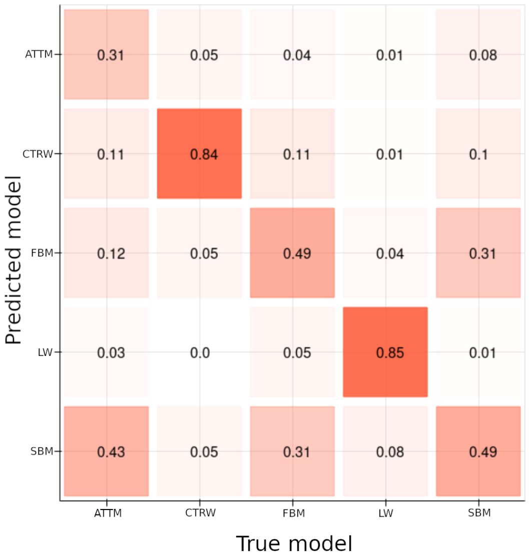


Fig. 6.9: F1-scores of noisy trajectories, with SNR = 1, by trajectory length for GASF/ResNet (a), GADF/ResNet (b), and ConvLSTM (c) for each model (ATTM, CTRW, FBM, LW, and SBM)



	ATTM	CTRW	FBM	LW	SBM
ATTM	0.31	0.05	0.04	0.01	0.08
CTRW	0.11	0.84	0.11	0.01	0.1
FBM	0.12	0.05	0.49	0.04	0.31
LW	0.03	0.0	0.05	0.85	0.01
SBM	0.43	0.05	0.31	0.08	0.49

Fig. 6.10: Results of the GASF/ResNet model under the andi interactive tool.

Team	F1-score	AUC	Model Type
eduN	0.499	0.82	RNN + Dense NN [14]
FCI	0,525	0,86	CNN [71, 131]
UPV-MAT	0.560	0.87	CNN + biLSTM[17]
GADF/ResNet	**0.581**	**0.89**	GADF/ResNet

Table 6.1: Performance comparison of GADF/ResNet model with best AnDi Challenge models at classification of the underlying diffusive regime in trajectory lengths 10 to 50.

6.5.2 Inference of the anomalous diffusion exponent

The anomalous diffusion exponent highly conditions how the diffusion occurs. In this section, we will analyze and compare the performance of our GASF and GADF ResNet Models to our ConvLSTM. In order to assess accuracy, we will use the *mean absolute error* (MAE) between the exponent used to generate each trajectory α_{truth} and the predicted exponent value α_{pred}. Given a data set containing N samples, the general MAE is defines as:

$$\text{MAE} = \frac{1}{\text{N}} \sum_{\text{j}=1}^{\text{N}} |\alpha_{\text{j,pred}} - \alpha_{\text{j,truth}} \tag{6.9}$$

As with the classification task, we will compare the performance of our GAF models with the ConvLSTM, but this time for the regression of the anomalous exponent α. Once again, GASF and GADF fed ResNet models performed the best for noisy trajectories with SNR = 1. For trajectories with SNR = 2, the performance of all five models tested is remarkably close, but the ConvLSTM provides better performance at shorter trajectory lengths (Figure 6.11). Still the difference in performance is quite marginal and we would favor the GASF and GADF ResNets for their ease of deployment and marginally better performance in noisier trajectories. When we consider trajectory length we have the same story. MAE is inversely proportional to the trajectory length with all diffusive regimes following the same trend of increased performance (lower MAE) with longer trajectory length across GASF/GADF ResNet models and the ConvLSTM (Figure 6.14)

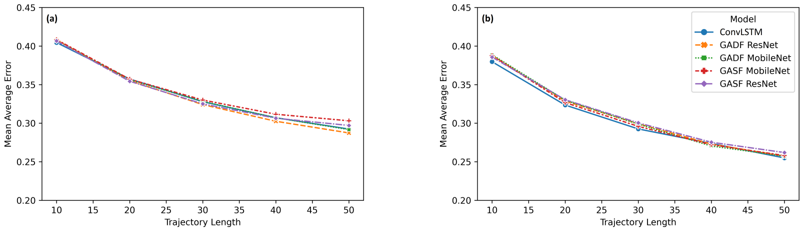


Fig. 6.11: Models' performance comparison for noisy trajectories with SNR = 1 (a) and SNR = 2 (b).

With regards to model performance as a function of the anomalous diffusion exponent alpha. We can see that all three models struggle are best able to infer the α of trajectories with $\alpha \in [0.5, 0.8]$ (Figure 6.12). We do not know what the reason for this is, but it is in opposition to the classification task, where all models struggled to classify trajectories that were roughly normal. This seems to indicate that the different diffusive regimes behave similarly at $\alpha \approx 1$, which is equivalent to increasing the amount of training data at $\alpha \approx 1$. This is because our models no longer have to account for which of the five models it is looking at in order to infer the α. Again in contrast to what we observed in the classification task, the worse performance was had at both extremes of each trajectories α domain. For instance MAE for CTRW was highest in all three models when

$\alpha \approx 0.1$ and $\alpha \approx 1.0$. Likewise SBM performed worst when it was at its most super and subdiffusive extremes (Figure 6.14). Again we believe that this is because diffusive regimes are at their most distinct (easiest to classify) at the extremes of their α domain. This in turn confuses our models as they must learn specifically to infer α for a specific underlying regime. This effect would be equivalent to decreasing the amount of training data.

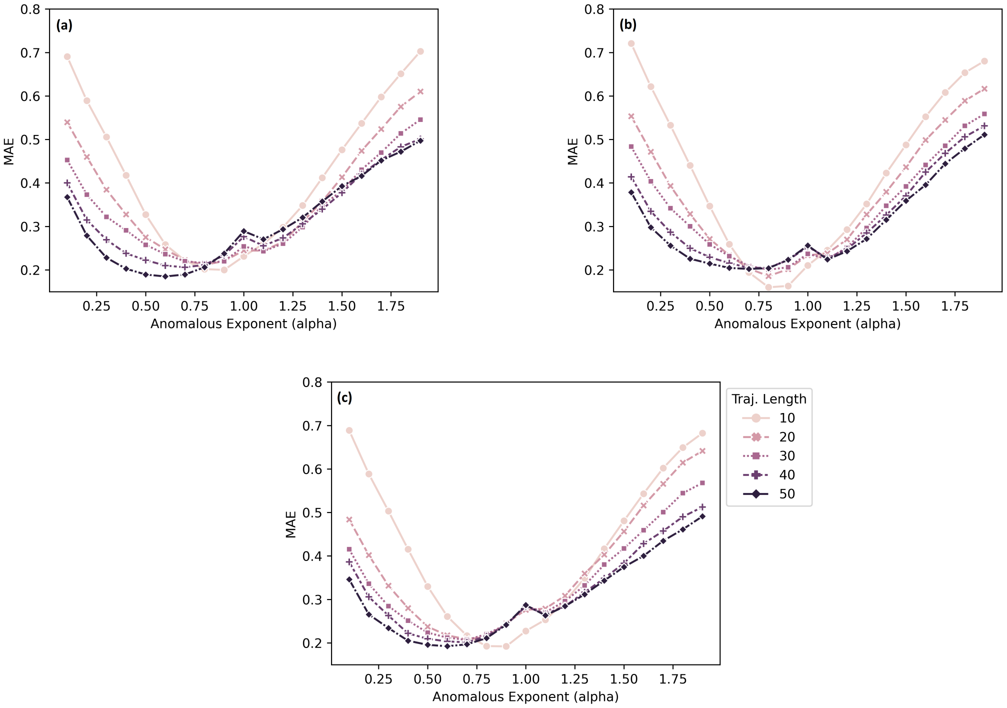


Fig. 6.12: MAE of GASF/ResNet (a), GADF/ResNet (b) and ConvLSTM (c) for different trajectory lengths and values of the anomalous exponent α on noisy trajectories with SNR $= 1$.

Team	**MAE**	
eduN	0,385	RNN + Dense NN [14]
FCI	0,369	CNN [71, 131]
UCL	0.367	feature engineering + NN [16]
UPV-MAT	**.326**	CNN + biLSTM[17]
GADF/ResNet	0.33	GADF fed ResNet

Table 6.2: Regression: performance comparison of GASF/ResNet model with best AnDi Challenge models.

Lastly we show a heat map of the inference results generated using the AnDi interactive tool in Figure 6.15. The heat map shows the distribution of the predicted α as a function of the ground truth. More heat maps, that predicted α as a function of true α as well as the underlying diffusive regime can be seen in Figure 6.16. Furthermore, Table 6.2 shows a summarized output of the AnDi interactive tool for the 3 best performing models in the AnDi challenge 2020 compared to our GADF ResNet. We can see that our GADF/ResNet outperforms the majority of them, except of UPV-MAT, but the difference in performance is small and we can say that both models behave

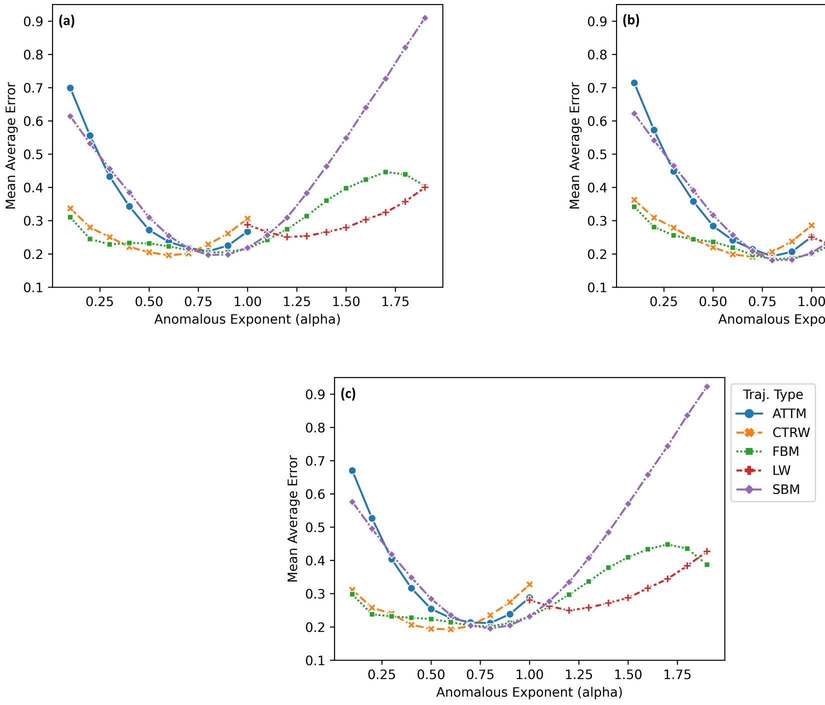


Fig. 6.13: MAE of GASF/ResNet (a), GADF/ResNet (b) and ConvLSTM (c) for different anomalous exponents α and (ATTM, CTRW, FBM, LW, and SBM) models of noisy trajectories with SNR = 1.

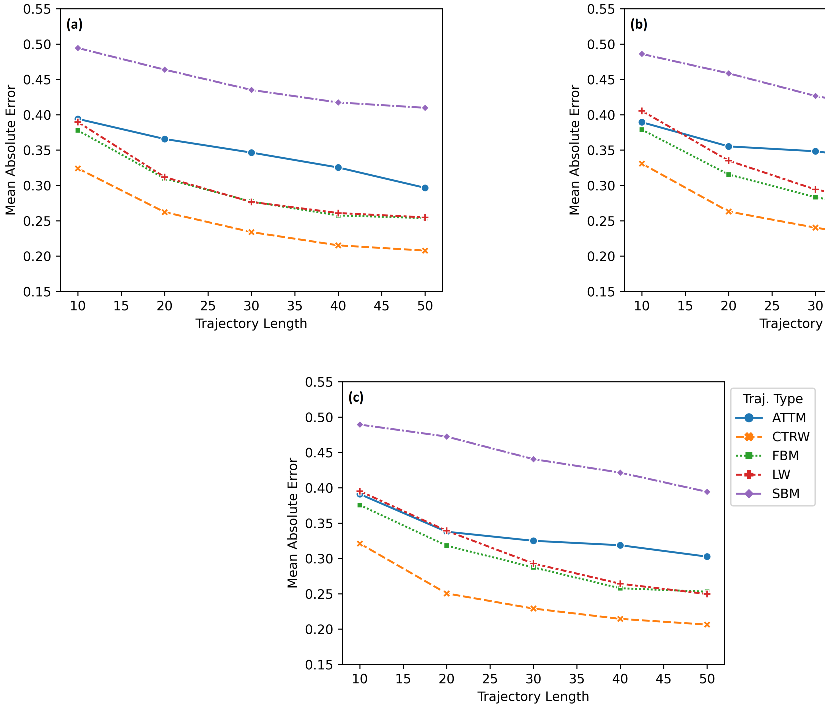


Fig. 6.14: MAE of GASF/ResNet (a), GADF/ResNet (b) and ConvLSTM (c) for different trajectory lengths of noisy trajectories with SNR = 1.

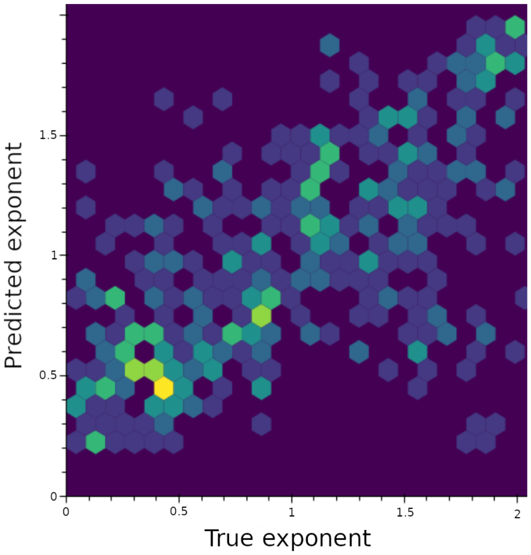


Fig. 6.15: Results of the GASF/ResNet model under the andi interactive tool for trajectory lengths 10 to 50.

roughly the same, with the GADF ResNet model being far easier to implement, as the ConvLSTM for inference of α is actually a compilation of 12 individually trained ConvLSTMs.

6.6 Conclusions

We set out with the objective of matching or improving the current methodology for the characterization of short and noisy diffusive trajectories, while increasing accessibility to ML methods by decreasing the difficulty of deployment. Our GASF and GADF ResNet Models outperformed the current state of the art by a wide margin in trajectory classification, and were negligibly worse than the best method for regression of the anomalous diffusion exponent α. These results, especially for classification, exceeded our hopes, particularly considering the ease of implementation. ResNet can be natively implemented by using Keras in Python, and the *pyts.image* package allows one to convert time series to Gramian matrices with a single function.

Diffusion is all around us, and perturbations to a normal diffusive regime, such as confining movement to a room, diffusion on a fractal, or even a breeze have the capability to make this diffusion anomalous. Traditionally, particularly in a more applied setting like ecology, the lack of tools and availability of data lead researchers to assume that movement was normal. The lens of diffusion provides an interdisciplinary framework for the characterization of movement, which should increase collaborations between traditionally insular fields. We hope that with the rise of flexible tractable ML models, such as our GASF/GADF fed ResNet, facilitate a new wave of applied interdisciplinary diffusive studies, which advance our knowledge of diffusion regardless of scale.

Acknowledgements

We thank M.A. García-March for helpful comments and discussions on the topic. NF is supported by the National University of Singapore through the Singapore International Graduate Student Award (SINGA) program. OGO and LS acknowledge funding from MINECO project, grant TIN2017-88476-C2-1-R. JAC acknowledges funding from grant PID2021-124618NB-C21 funded by

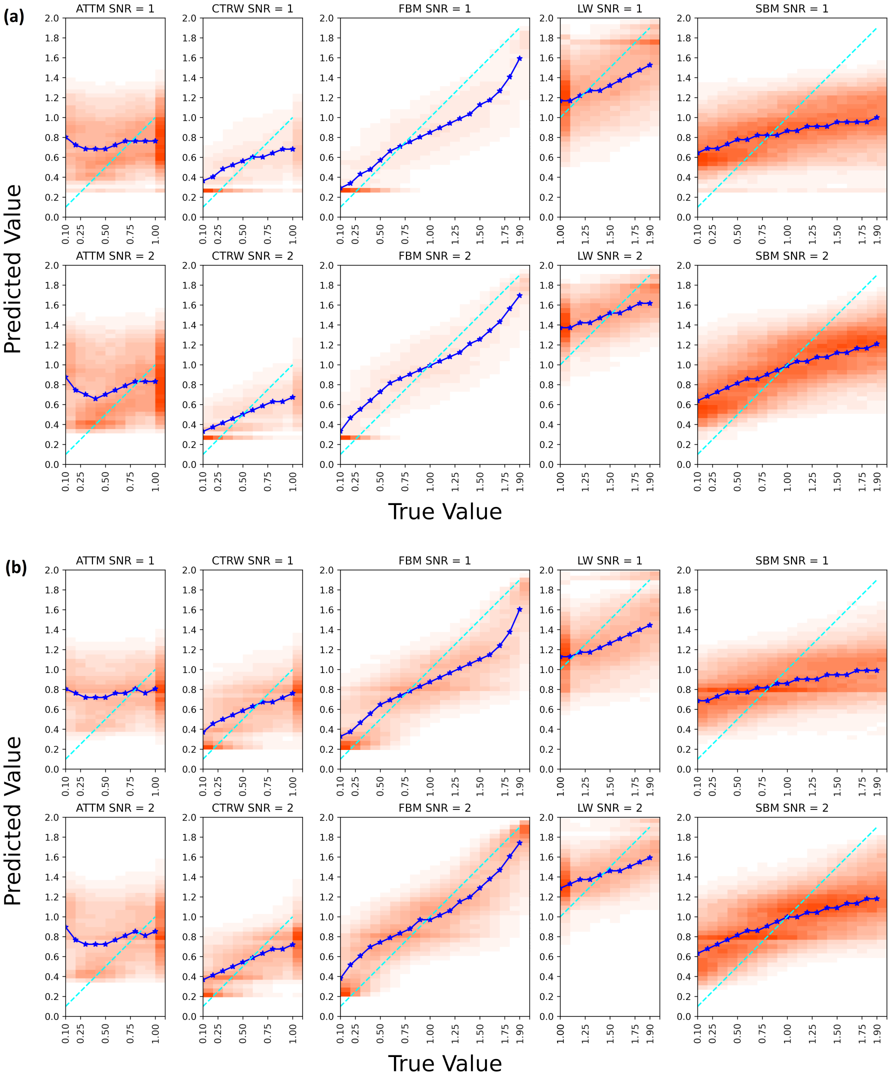


Fig. 6.16: Plots (a) and (b) show the results for the GASF and GADF ResNet models in the inference of the anomalous diffusion exponent α. The blue star line represents the median prediction for each of the true values and the cyan dashed line represents the true value line.

MCIN/AEI/ 10.13039/501100011033 and by "ERDF A way of making Europe", by the "European Union".

7 Journal article (iii)

Who in the world am I?
Ah, that's the great puzzle.

Alice, from Alice in Wonderland.

Inferring the fractional nature of Wu Baleanu trajectories

Inferring the fractional nature of Wu Baleanu trajectories

Inferring the fractional nature of Wu Baleanu trajectories. *Nonlinear Dynamics.* Accepted for publication.

Conejero, J.A.[2], Garibo-i-Orts, O.[1,2], Lizama, C.[3]

1 GRID - Grupo de Investigación en Ciencia de Datos, Valencian International University - VIU, Carrer Pintor Sorolla 21, 46002 València, Spain.
2 Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain.
3 Departamento de Matemática y Ciencia de la Computación Universidad de Santiago de Chile, Las Sophoras 173, Estación Central, Santiago, Chile.

Abstract. We infer the parameters of fractional discrete Wu-Baleanu time series by using machine learning architectures based on recurrent neural networks. Our results shed light on how clearly one can determine that a given trajectory comes from a specific fractional discrete dynamical system by estimating the fractional exponent and the scaling factor. With this example, we also show how machine learning methods can be incorporated into the study of fractional dynamical systems.

7.1 Introduction

The logistic equation introduced by May [132] models the behavior of a population that grows exponentially, but some constraints of the environment limit this growth. We can express it as

$$v(n+1) = \eta v(n)(1 - v(n)), \qquad \text{for } n \in \mathbb{N}_0, \tag{7.1}$$

where $v(0) \in [0,1]$ and $\eta \in \mathbb{R}$. This equation provides the simplest example of a one-parameter nonlinear dynamical system with nontrivial dynamics. It is very well-known that if $0 \leq \eta \leq 4$, we have a well-defined dynamical system on $[0,1]$. For $\eta > 4$, we still can have a discrete dynamical system, but this will only be defined on the complementary of a particular Cantor set in $[0,1]$; see for instance [133].

In order to incorporate some difference operator that we can later extend naturally with a discrete fractional derivative, we can transform the logistic equation by applying the change of variable $v(n) = \frac{\eta}{\eta-1}u(n)$. In this way, instead of having a formula for computing the term $u(n+1)$

by recurrence, we express that the forward Euler operator Δ is equal to the nonlinear right term of the logistic, that is

$$\Delta u(n) := u(n+1) - u(n). \tag{7.2}$$

So, we obtain the logistic equation of parameter $\mu := \eta - 1$ with the initial condition rescaled by a factor $\frac{\mu+1}{\mu}$. More precisely, we have

$$\Delta u(n) = \mu u(n)(1 - u(n)), \qquad u(0) = \frac{\mu+1}{\mu} v(0). \tag{7.3}$$

Wu and Baleanu [134] considered a fractional version of the dynamical system generated by (7.2) replacing the Euler operator by the left Caputo discrete difference operator Δ^{α}. It is interesting to observe that with the given definition of Δ^{α}, the fractional version of the logistic equation adopts a convolutional form and reads as follows

$$u(n) = u(0) + \frac{\mu}{\Gamma(\nu)} \sum_{j=1}^{n} \frac{\Gamma(n-j+\nu)}{\Gamma(n-j+1)} u(j-1)(1-u(j-1)). \tag{7.4}$$

Given an arbitrary condition $u(0) \in [0,1]$, the trajectory $\{u(n)\}_{n=1}^{N}$ obtained with (7.4) will be called a Wu-Baleanu trajectory. It is worth to mention that for $\nu = 1$ and $n = 1$ we have $\Delta u(0) = \mu u(0)(1 - u(0))$, recovering (7.2) for $n = 0$.

There is an alternative way of deducing this equation 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 kernels [135]. This representation gives an interpretation of the fractional version of the logistic equation as the one that incorporates a memory kernel in terms of a discrete parameter, thus incorporating a different measure of the trajectories.

Fixing an initial condition and a scaling factor, we can generate Feigenbaum diagrams in order to illustrate the dynamics of this dynamical system in terms of the parameter μ, see Figures 7.1 and 7.2.

Anomalous diffusion trajectories $\{u(n)\}_{n=1}^{N}$ are those whose average width or variance, computed as the Mean Square Displacement (MSD) $\langle u(n) - u(0) \rangle$ do not grow linearly with respect to n, that is $\langle u^2 \rangle \approx n^{\alpha}$, with $\alpha \neq 1$. The exponent α is known as the anomalous diffusion exponent. Examples of models generating anomalous diffusion trajectories are: Annealed Transient Time Motion (ATTM) [33], Continuous Time Random Walk (CTRW) [31], Fractional Brownian Motion (FBM) [37, 38], Lévy Walks (LW) [10, 32], and Scaled Brownian Motion (SBM) [39].

Recently, within the frame of the Andi Challenge [117], machine learning methods, alone or combined with some statistical measures, have demonstrated their efficiency in (i) classifying anomalous diffusion noisy trajectories according to one of the previous five generative models and (ii) inferring the anomalous diffusion exponent. We refer the reader to [10] and some subsequent works in which some of these models were fully developed [14, 16, 17]; see also [19, 66, 136]. It is also worth mentioning the recent interest in incorporating machine learning methods and intelligent algorithms in the study of formal mathematical problems. We can find some examples of this approach incorporating these techniques for the solution of nonlinear models, such as artificial neural networks, swarm optimization, and active-set algorithms [137], or neuro-swarming heuristics [138]

Therefore, we wonder if this approach lets us infer some fractional-related trajectories' characteristics. In this work, we study if we can infer the μ parameter and the scaling factor ν of Wu-Baleanu trajectories. We have chosen an architecture based on recurrent neural networks (RNN), the same that provided the best results in inferring the exponent α of one-dimensional trajectories in the Andi Challenge [10, 17], for trying to infer these parameters and measuring up to which point there is a straightforward relation any given trajectory of this type and the corresponding parameters μ and ν involved in generating it. To the best of our knowledge, this paper is the first to seek this type of approach. In Section 7.2, we give some details of the model architecture, revisiting some basic fundamentals of our machine learning models. We set the training, validation, and test data sets, as long as the results, in Section 7.3. Finally, we draw some conclusions in Section 7.4.

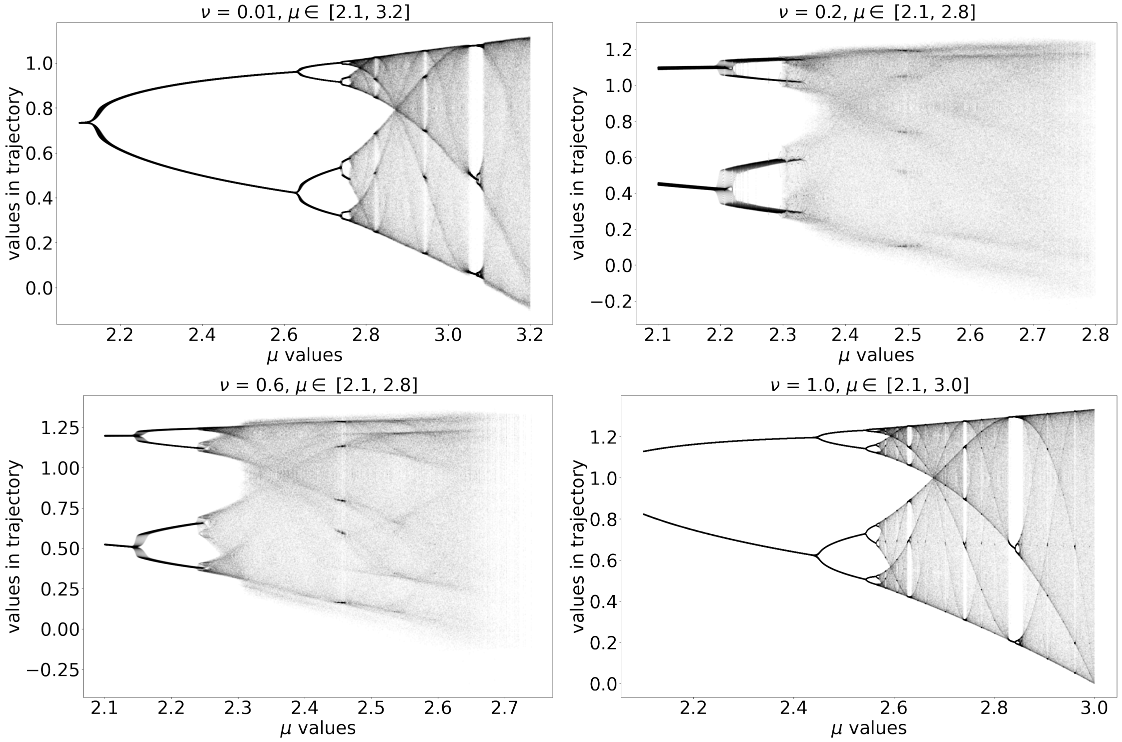


Fig. 7.1: Feigenbaum plots for the dynamical system given by (7.4) for $u_0 = 0.3$ and $\nu = 0.01$ (top left), $\nu = 0.2$ (top right), $\nu = 0.6$ (bottom left), and $\nu = 1$ (bottom right), For each value μ, we compute 200 terms of the sequence, and we plot the last 100 values.

7.2 Architecture of the method

We propose the architecture shown in Figure 7.3 to infer the parameters μ and ν of a given trajectory are introduced as input. We will consider trajectories of length between 10 and 50, which are the most frequent in experiments and the hardest to be classified [10]. Such architecture has been successfully applied for analyzing trajectories [17] and time series [20]. It consists of three parts:

1. First, we have two convolutional layers that permit the extraction of spatial features from the trajectories. The first convolutional layer is set with 32 filters and a sliding window (kernel) of size 5, which slides through each trajectory extracting spatial features from them. The second convolutional layer has 64 filters to extract higher-level features.
2. Second, the output of the convolutional layers feeds three stacked bidirectional LSTMs layers that permit learning the sequential information. After each of these layers, we include a dropout layer of the 10% neurons to avoid over-fitting. We tested several dropout levels, from 5% to 20%, being 10% the one with the best performance.
3. Finally, we use two fully connected dense layers: the first one with 20 units and the second one with 1 or 2 units. This last choice depends if we want to predict a single parameter or both of them at the same time.

Let us briefly describe each part of the model:

7.2.1 Convolutional neural networks (CNN)

Convolutional neural networks preserve the spatial structure of data. They do so by connecting a patch (or section) from data to single neurons, so every neuron learns the properties from this

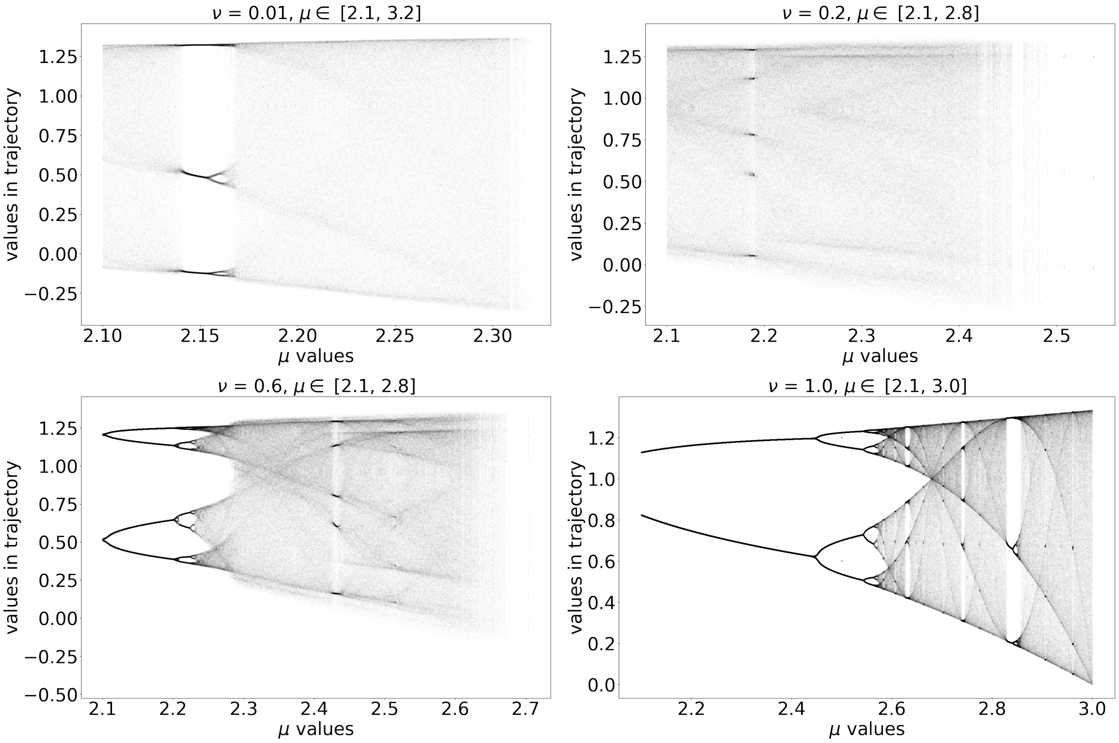


Fig. 7.2: Feigenbaum plots for the dynamical system given by (7.4) for $u_0 = 0.8$ and $\nu = 0.01$ (top left), $\nu = 0.2$ (top right), $\nu = 0.6$ (bottom left), and $\nu = 1$ (bottom right), For each value μ, we compute 200 terms of the sequence, and we plot the last 100 values.

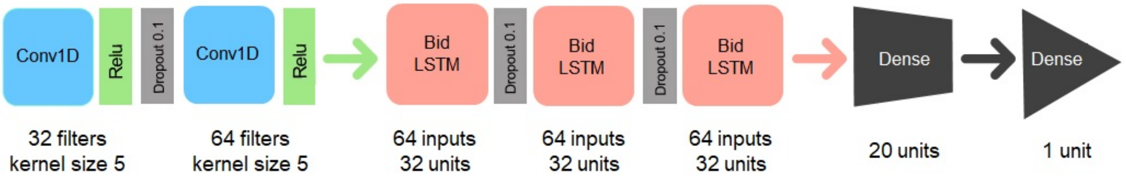


Fig. 7.3: Machine learning used for inferring the generating μ and ν parameters of Wu-Baleanu trajectories.

single patch, whose size is defined by the kernel size (5 in our model). By doing so, spatially close portions of data are likely to be related and correlated to each other since only a small region of the input data influences the output from each neuron [1, 2]. The patch is slid across the input sequence, and each time we slide it, we have a new output neuron in the following layer. This lets us consider the spatial structure inherent to the input sequence [3, 4]. Through these layers, we are able to learn trajectory features by weighting the connections between the patches and the neurons so that particular features can be extracted by each patch. By using multiple filters (32 and 64 in our case) the CNN layers are extracting multiple different features (linear and non-linear), that feed our LSTM layers.

7.2.2 Recurrent Neural Networks (RNN)

Sequential information can be decomposed in single-time steps, such as words or characters in language, notes in music, codons in DNA sequences, etc. So, if one considers sequential data it is very likely that the output at a later time step will depend on the inputs at prior time steps. In

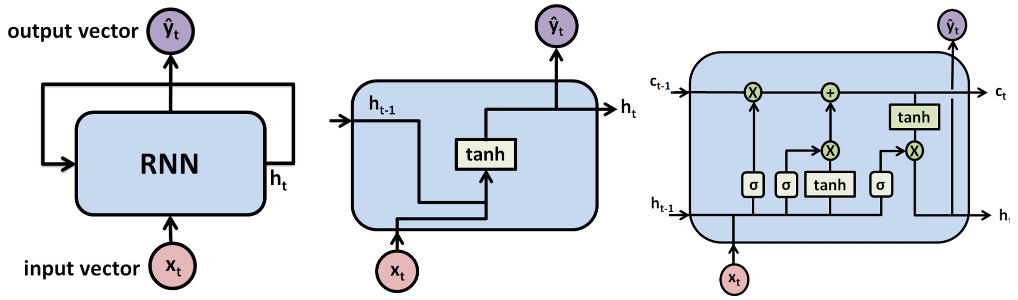


Fig. 7.4: Basic representation of a general RNN (left). An RNN with an inner tanh activation function (center). A scheme of an LSTM layer (right).

practice, we need to relate the information from a particular time step also with prior time steps and pass this information to future times.

Recurrent neural networks (RNN) address this problem by adding an internal memory or cell state, denoted by h, which is passed from the time t to the time $t+1$, that is from h_t to h_{t+1}. This recurrent relation is capturing some notion of memory of what the sequence looks like. Therefore, the RNN output is not only a function of the input at a particular time step but also a function of the past memory of the cell state. In other words, the output $\overline{y_t} = f(x_t, h_{t-1})$, depends on the current input x_t and the previous inputs to the RNN h_{t-1}, as it can be seen Figure 7.4.

An RNN adapts the internal hidden state (or memory state) h_t through the result of multiplying two weight matrices W_{hh} and W_{xh} to the previous cell state h_{t-1} and the current input x_t, respectively. The weight matrix W_{hh} is modified at each time step to let the cell learn how to fit the desired output, and W_{xh} is the weight matrix that modules the contribution of the input at each time step to the learning process. The result is passed to an activation function tanh that modifies the current state at each time step, i.e. $h_t = \tanh\left(W_{hh}^T h_{t-1} + W_{xh}^T x_t\right)$.

The problem with RNNs arises when dealing with long sequences since composing multiple tanh functions entails that the hidden state tends to extinguish by reaching values very close or equal to zero. In practice, this means that only recent cell states will modify the current cell state or, in other words, that RNNs have short-term memory.

7.2.3 Long short-term memory (LSTM)

Long short-term memory (LSTM) [5, 6] amend the aforementioned short-term memory problem implicit to RNN by including gated cells that allow them to maintain long-term dependencies in the data and to track information across multiple time steps. This improves the sequential data modeling. LSTM structure is shown in Figure 7.4 where σ and tanh stand for the sigmoid and the hyperbolic tangent activation functions The circles in red represent matrix multiplication and additions. An LSTM incorporates a new cell state channel c which can be seen as a transportation band where the info is selectively updated by the new gates and is independent of the previously defined hidden state h and, therefore, independent of what is outputted in the form of hidden state or current time step out.

One LSTM cell's composition can be seen in Figure 7.4 (right), and the gates are used to control the flow of information as follows:

- The first sigmoid gate decides what information is kept or rid of. Since the sigmoid output ranges from 0 to 1, this can be seen as a switch that modulates how much information from the previous state has to be kept.
- The second gate, consisting of a sigmoid and a tanh functions store relevant information to the newly added cell state channel (c).

- Then, the outputs of the two previous gates are used to update the cell state (c) selectively.
- And the last sigmoid and tanh functions produce two different outputs; the new cell state (c), which is forwarded to the next LSTM cell, and the current time step output, which is a filtered version of the cell hidden state (h).

Further details about LSTM functioning and implementation can be found in [7, 8].

7.3 A general model for inferring μ and ν parameters

We have built three independent data sets to infer both μ and ν simultaneously. The train, validation, and test data sets have been built with the following parameters:

- $\mu \in [2, 3.2]$ 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 visit the μ range with higher accuracy, in order to capture the chaotic dynamics that appears at some regions, see Figures 7.1 and 7.2. We iterate over the values of μ and ν for building the aforementioned data sets. At each iteration, per each combination of μ and ν values we randomly select 5 length value N and 5 different values of u_0, one in each one of these intervals [0.0, 0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8] and (0.8, 1.0], thus producing 5 trajectories of different lengths. When computing the trajectories, if we attain a value lower than -0.5 or greater than 2.0 in the trajectory, we stop the trajectory generation and save the trajectory as it is, provided it has a length greater than 10. The whole pool of trajectories is split into training (65%), validation (15%), and test (20%). As a result of this procedure, we get a training data set containing 618199 trajectories, a validation data set of 142822 trajectories, and a test data set with 190334 trajectories. The data sets can be found in the supplementary material.

In all data sets, we pad each trajectory with 0's at its beginning to make them of a fixed length equal to 50. This permits homogenizing the lengths and feeding the first convolutional layer of our proposed architecture, see [139, Ch. 5 & Ch. 9]. We used an early stopping callback with a patience value of 20, which in practice means that the model stops training when validation mean average error (MAE) does not improve after 20 consecutive epochs. 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 2 hours, running up to 23 epochs.

First, we provide a description of the MAE distribution in terms of the true value to be predicted in Figure 7.5. The diagonal spot along the diagonal represents that the model predicts very well the parameters μ and ν. On the one hand, when inferring the parameter μ, the best results are given for medium values of μ, between 2.24 and 2.96; on the other hand, the results behave more homogeneously for ν.

We point out that the results in Figure 7.5 do not shed light on the importance of the trajectory length to improve the accuracy of the predictions. It is expected that the longer the trajectories are, the lower the MAE is. In order to check it, we compare the MAE results for the shortest trajectories, lengths between 10 and 19, and the longest ones, lengths between 40 and 50, in Figure 7.6.

Here, it is not easy to appreciate at first sight; however, looking at Table 7.1, we can see that it really holds. We can see that, except in the last case, as we increase the trajectory length, the results improve since the parameters can be better identified. In all cases, we are in MAEs of the order of 10^{-2}, that is the same order of magnitude of the ν parameter discretization, and one order less than the μ one. This justifies our choice of a thinner discretization for μ with respect to ν.

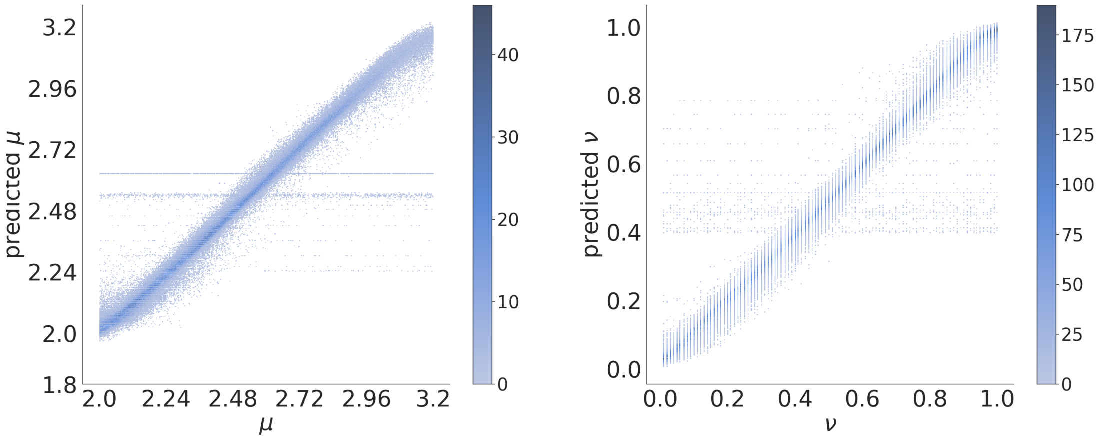


Fig. 7.5: Truth vs predicted values of μ and ν in the validation data set.

Length	**MAE (μ)**	**MAE (ν)**
10-19	0.0276	0.0211
20-29	0.0234	0.0173
30-39	0.0233	0.0164
40-50	0.0262	0.0186
All	0.0253	0.0186

Table 7.1: μ and ν MAE in the test data set.

In order to look for more insightful descriptions of the MAE, for each truth value of μ and ν we have represented the quartiles Q_1, Q_2, and Q_3 of the MAE error distribution in Figure 7.7. We can see that the predictions for μ are less accurate at the extremes than in the inner region, as we have already noticed in Figure 7.5. However, we see here more clearly, that the models are less accurate close to the extreme values of $\nu = 0$ and 1 due to the strong connection existing between both scaling values. We provide a comparative of these MAE distributions for short and long trajectories in Figure 7.8.

Due to the fractional nature of equation (7.4), the model has a memory component that is strongly dependent on the initial condition $u(0)$. We show boxplots of the MAE distribution in terms of $u(0)$ in Figure 7.9. We can see that initial conditions are more influential for μ predictions since boxes (green) and whiskers (grey) are higher than for ν. Despite of this, in both cases, the results are slightly worse for initial conditions closer to 1, due also to the nonlinear term of the logistic terms of (7.4). The same conclusion can be extracted when evaluating trajectories by length, as shown in Figure 7.10.

7.4 Conclusions

The success of machine learning and deep learning models has arrived in almost all scientific fields. The development of mathematical proofs and arguments seems to be one of the most difficult challenges. Nevertheless, some barriers have already fallen with the discovery of new multiplication algorithms [140].

Machine learning methods can also help us in modeling tasks and in the search and fitting of parameters. In this line, we have shown these methods permit us to infer the fractional nature of a given trajectory. In particular, we have seen that such a model permits us to elucidate if, given

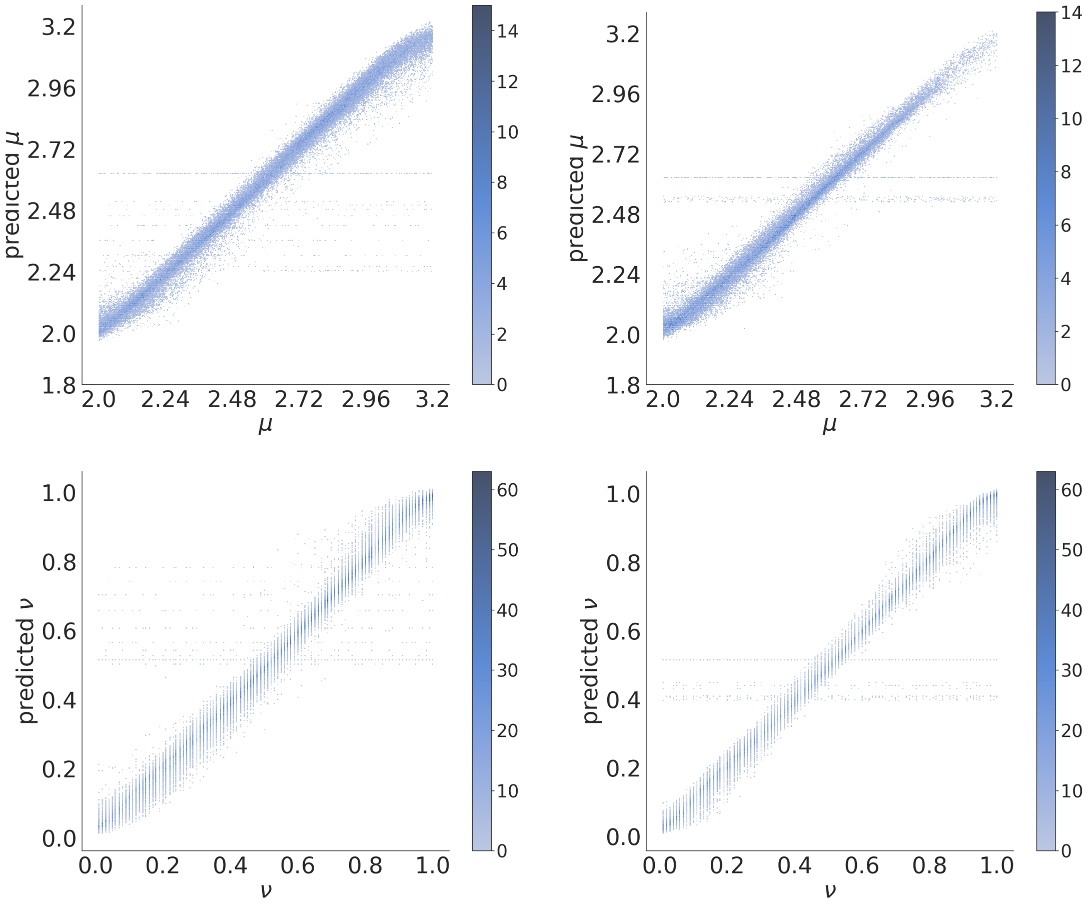


Fig. 7.6: Truth vs predicted values for μ and ν in the validation data sets for trajectory lengths [10-19] (left) and [40-50] (right).

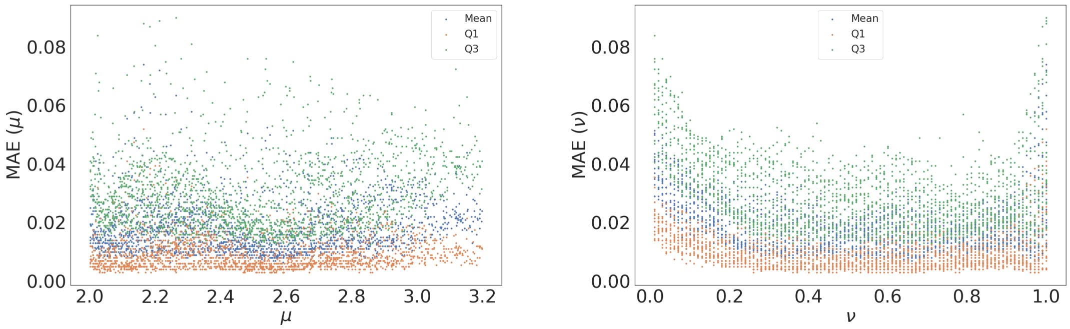


Fig. 7.7: Quartiles Q1 (red), Q2 (blue), and Q3 (green) of the MAE distribution on the evaluation data for μ (left) and ν (right).

a set of trajectories, we can propose a fractional model based on the logistic equation that would represent the underlying process with reliability. Moreover, with reasonable use of resources, we can tune the model in order to estimate the parameter of the model μ and the scaling factor ν of the fractional discretization, within a similar order of magnitude.

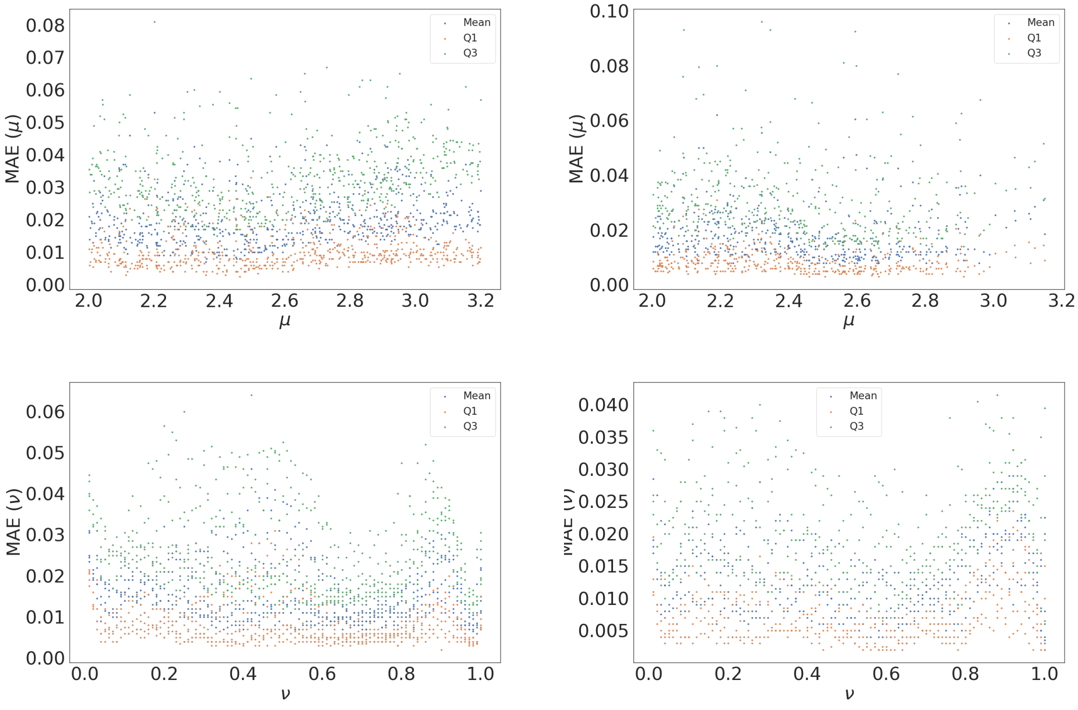


Fig. 7.8: Quartiles Q1 (red), Q2 (blue), and Q3 (green) of the MAE distribution on the evaluation data set associated with every single value of μ and ν for trajectory lengths [10-19] (left) and [40-50] (right).

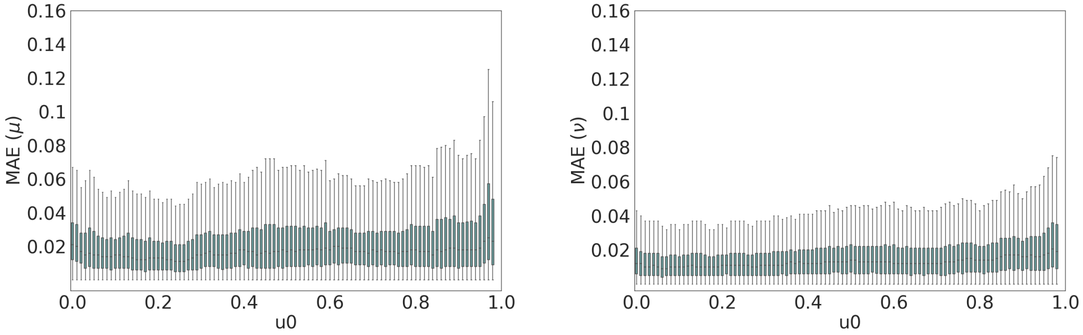


Fig. 7.9: MAE of μ (left) and ν (right) on the evaluation data set as a function of u_0. The darkest region coincides with the boxes, the medium grey stands for the whiskers and the light grey for the outliers.

We expect that this study will foster the incorporation of machine learning tools into the study of dynamical systems and the modeling of real-life problems.

Acknowledgements

We thank M.A. GarcÃa-March for helpful comments and discussions on the topic. JAC acknowledges funding from grant PID2021-124618NB-C21 funded by MCIN/AEI/ 10.13039 /501100011033 and by "ERDF A way of making Europe", by the "European Union".

We also thank funding for the open access charges from CRUE-Universitat Politècnica de València.

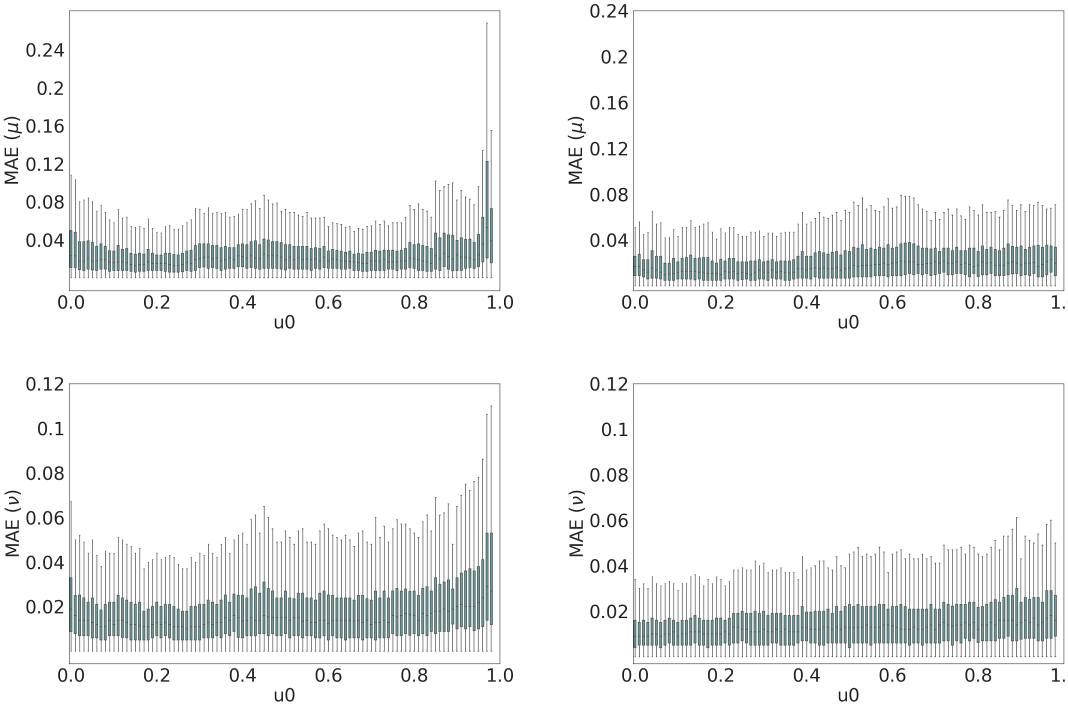


Fig. 7.10: MAE of μ (up) and ν (down) on the evaluation data set as a function of u_0 for trajectory lengths [10-19] (left) and [40-50] (right). The darkest region coincides with the boxes, the medium grey stands for the whiskers and the light grey for the outliers.

Ethics declarations

The authors declare no conflict of interest.

8 Journal article (iv)

I remember that I am here not because of the path that lies before me,
but because of the path that lies behind me.

Morpheus, The Matrix Reloaded.

Recovering discrete delayed fractional equations from trajectories

Recovering discrete delayed fractional equations from trajectories

Conejero, J.A.[2], Garibo-i-Orts, O.[1,2], Lizama, C.[3]

1 GRID - Grupo de Investigación en Ciencia de Datos, Valencian International University - VIU, Carrer Pintor Sorolla 21, 46002 València, Spain.
2 Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain.
3 Departamento de Matemática y Ciencia de la Computación Universidad de Santiago de Chile, Las Sophoras 173, Estación Central, Santiago, Chile.

Abstract. 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.

8.1 Introduction

Discrete Fractional Calculus (DFC) has recently attracted increasing attention from a growing number of researchers. In the last years, several papers concerning this theory have appeared in the literature and they have been essential for building its theoretical foundations.

In 1956, Kutter [141] mentioned for the first time differences of fractional order. In 1974, Diaz and Osler [142] introduced a discrete fractional difference operator defined as an infinite series. In 1988, Grey and Zhang [143] developed a fractional calculus for the discrete ∇ (backward) difference operator. Miller and Ross [144] 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 [145] 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 [134]; see also [146] 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

$$x(n) = x(0) + \frac{\mu}{\Gamma(\nu)} \sum_{j=1}^{n} \frac{\Gamma(n-j+\nu)}{\Gamma(n-j+1)} x(j-1)(1-x(j-1)), \tag{8.1}$$

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 [135]. Wu and Baleanu later introduced a two-dimensional dynamical system including a delay term in [147], that reads as

$$\begin{aligned} x(n) &= x(0) + \frac{\mu}{\Gamma(\nu)} \sum_{j=1}^{n} \frac{\Gamma(n-j+\nu)}{\Gamma(n-j+1)} x(j-1)(1-y(j-1)) \\ y(n) &= x(n-1). \end{aligned} \tag{8.2}$$

This delayed fractional discrete dynamical system also presents chaos as it can be noticed in the Feigenbaum diagrams computed in terms of the parameter μ. As a matter of fact, in Figures 8.1 8.2, we show the Feigenbaum diagrams computed for the initial conditions $x(0) = y(0) = 0.3$ and $x(0) = y(0) = 0.8$.

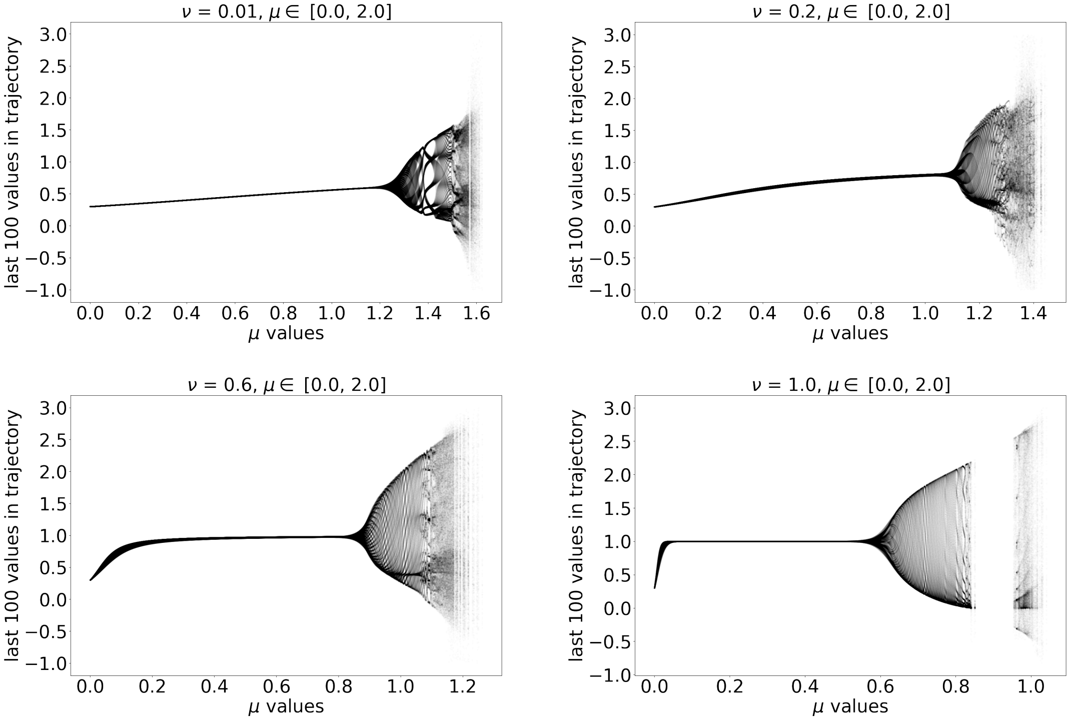


Fig. 8.1: Feigenbaum plots for the dynamical system given by (8.1) for $x(0) = y(0) = 0.3$ and $\nu = 0.01$ (top left), $\nu = 0.2$ (top right), $\nu = 0.6$ (bottom left), and $\nu = 1$ (bottom right), For each value μ, we compute 200 terms of the sequence, and we plot the last 100 values.

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 [117]. Examples of models generating those types of trajectories are Annealed Transient Time Motion (ATTM) [33], Continuous Time Random Walk (CTRW) [31], Fractional Brownian Motion (FBM) [37, 38], Lévy Walks (LW) [10, 32], and Scaled Brownian Motion (SBM) [39].

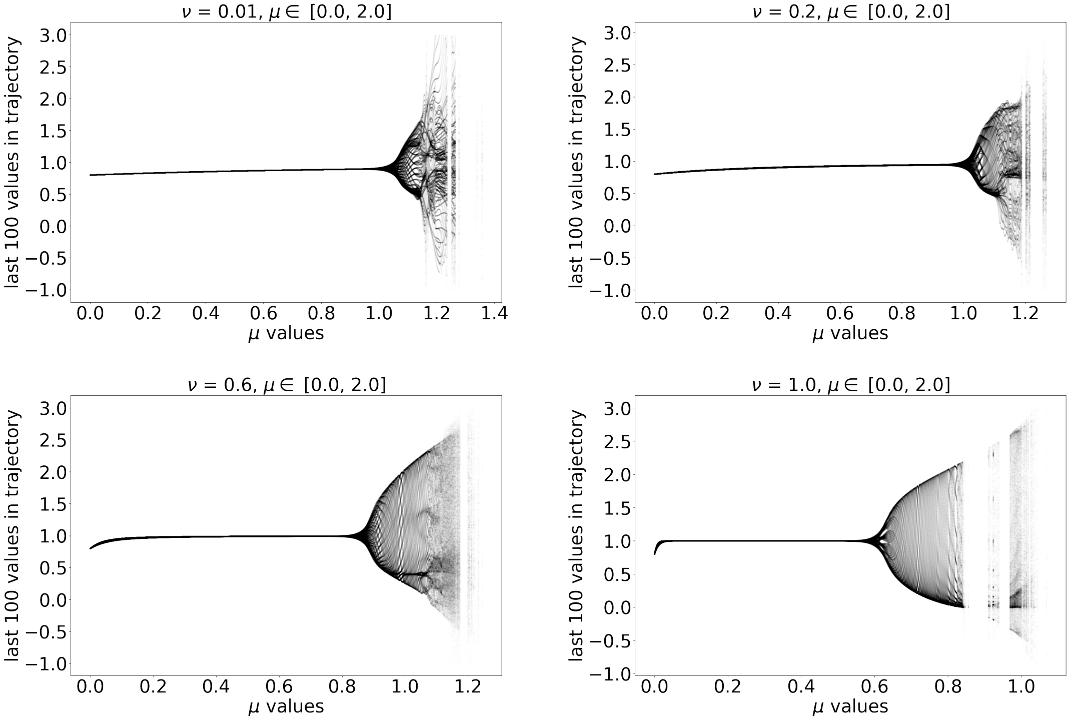


Fig. 8.2: Feigenbaum plots for the dynamical system given by (8.1) for $x(0) = y(0) = 0.8$ and $\nu = 0.01$ (top left), $\nu = 0.2$ (top right), $\nu = 0.6$ (bottom left), and $\nu = 1$ (bottom right), For each value μ, we compute 200 terms of the sequence, and we plot the last 100 values.

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 [14, 16, 17]. We refer the reader for a full comparison between these models to [10].

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 [137, 138] or the recent success of discovering new multiplication algorithms [140].

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 [148–152]. There is not a universal definition of fractional derivative [153] and in some cases, it is not straightforward to know which definition would be better to choose [154].

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 [9]. 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 (8.2). We also analyze if we can discriminate if trajectories contained a delayed component by comparing trajectories generated by the aforementioned systems described in (8.2) and (8.1).

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 [17] and for predicting the number of new COVID19 cases [20]. 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 8.2, 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 8.3, and we draw some conclusions in Section 8.4.

8.2 Methodology

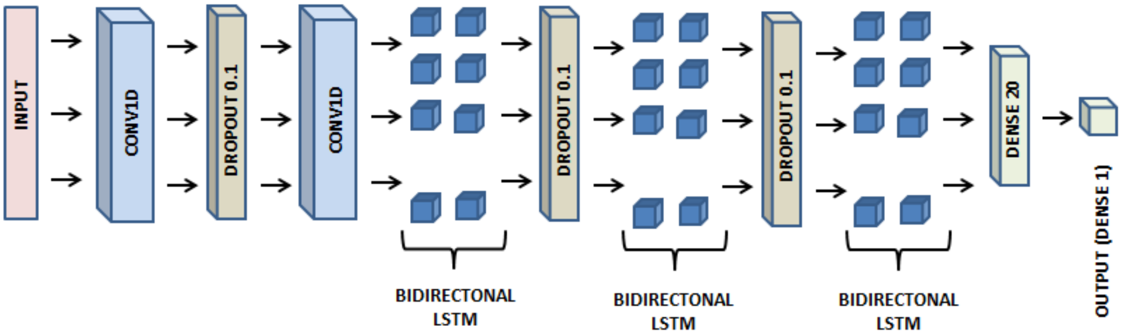


Fig. 8.3: Machine learning architecture used for inferring the generating μ and ν parameters of the delayed logistic model trajectories.

We start by describing our network architecture we depict in Figure 8.3 and that consists of 3 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 [3, 4].

   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 [3, 4]. The different number of filters returns features at different scales while retaining the spatial structure.

2. Secondly, the features feed 3 stacked bidirectional Long Short Term Memory (LSTM) of 64 inputs and 32 units each one [5, 6]. 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 8.1 and 8.2), the resolution of μ is increased by 10 times respect to the one of ν. We summarize the data generation procedure in the algorithm **??**.

Algorithm 1 Data sets creation algorithm

```
for <every $\mu$> do
    for <every $\nu$> 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 $\leq$ 0.20 then
                    <Save test trajectory>
                else
                    if index $\leq$ 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
```

As described in Algorithm 1, for every pair of (μ, ν) values, we generate 5 trajectories or random length (between 10 and 50), each one with a random initial condition chosen in each one of these 5 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 zeroes to reach the maximum length (50). In other words, zeroes are inserted to the left of the trajectory to make all of them have a length equal to 50, see [139, Ch. 5 & 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 hours, running up to 24 epochs.

8.3 Results

In Figure 8.4, 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.

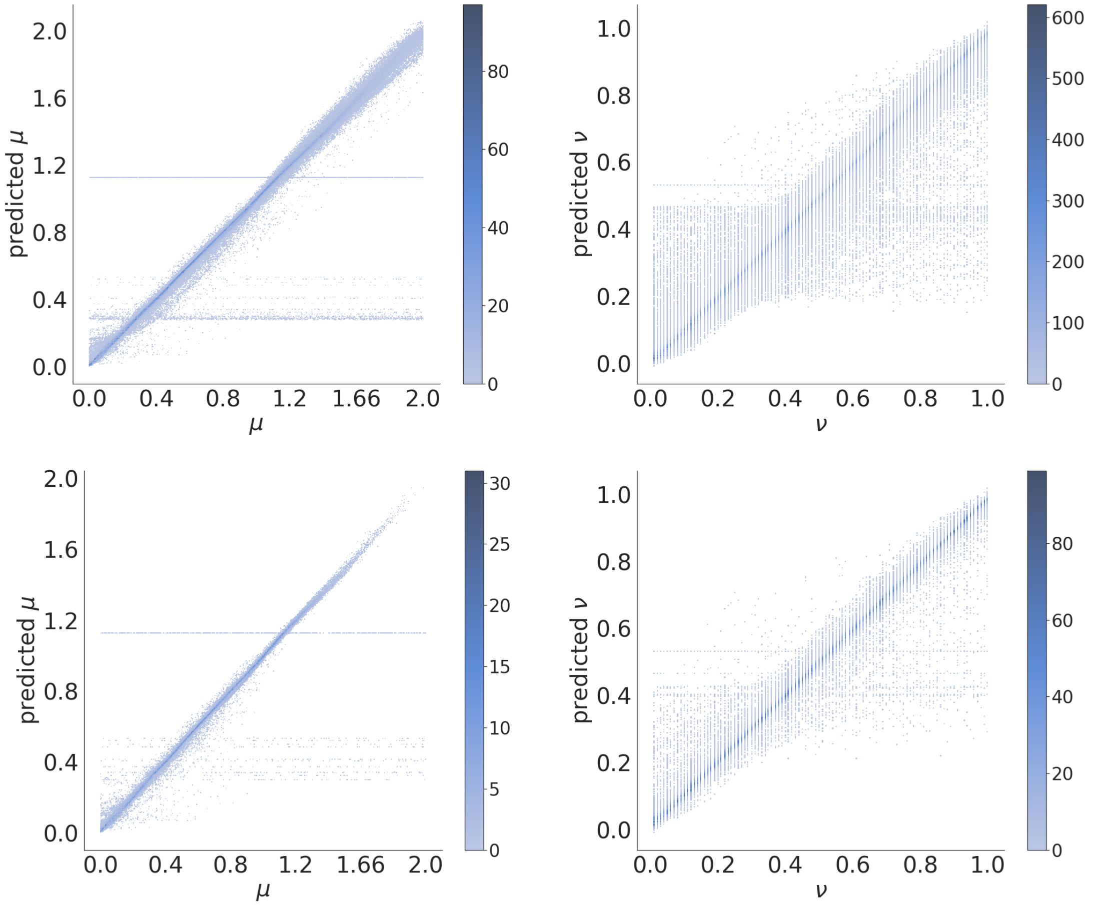


Fig. 8.4: Truth vs predicted values of μ and ν in the validation data set (top) and for trajectories of length between 40 and 50 (bottom).

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

Table 8.1: μ and ν MAE's in the test data set.

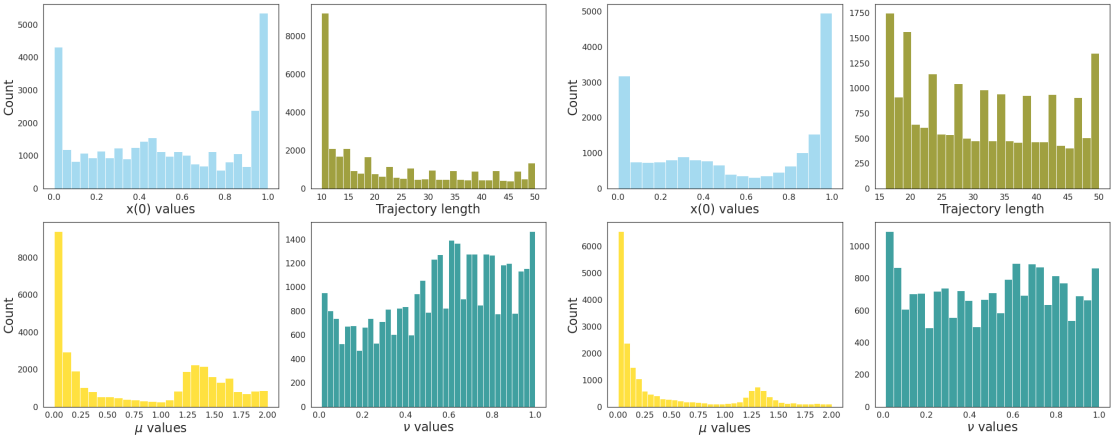


Fig. 8.5: Histograms of $x(0)$, trajectory lengths, μ, and ν values for trajectories with ν MAE higher than 0.05 (left) and for trajectories with ν MAE higher than 0.05 and length greater than 15 (right).

In Table 8.1, 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}) 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 20000 thousand for length equal to 10 but around 6000 for lengths between 40 and 50. As it was also noticed in [9] 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 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 8.5, 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 8.5 (right), we show the same histograms but just for trajectories of length greater than 15. Here, we see that despite 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 8.6, 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.

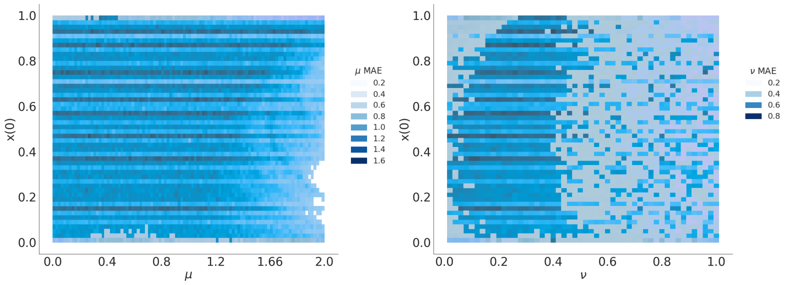


Fig. 8.6: μ (left) and ν (right) values vs $x(0)$ by μ and ν MAE respectively.

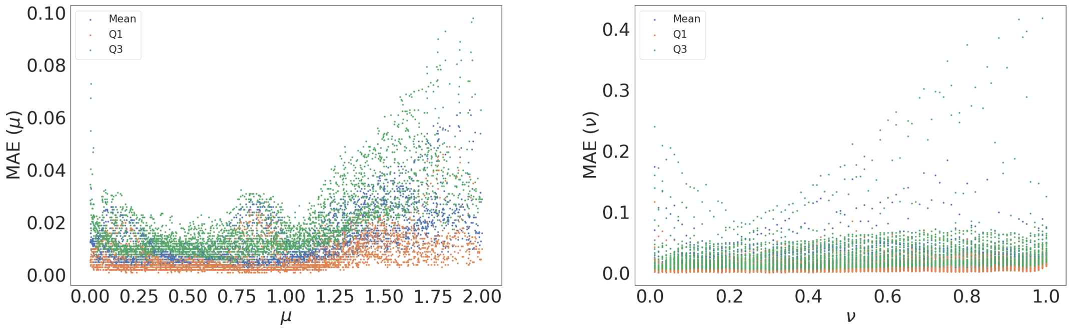


Fig. 8.7: Quartiles Q1 (red), Q2 (blue), and Q3 (green) of the MAE distribution on the evaluation data for μ (left) and ν (right).

We also study the MAE distribution for every μ and ν truth value. In Figure 8.7, 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 8.6. 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 8.1 and 8.2, and from the comparison of predicted and truth values of μ and ν in Figure 8.4.

Due to the fractional nature of equation (8.2), the model has a memory component that is strongly dependent on the initial condition $x(0)$. In Figure 8.8, 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 (grey) 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 (8.2).

Finally, we wonder if machine learning methods are able to determine which trajectories are generated from (8.1) and which ones are given by (8.2). 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 [9]. This second data set was also generated following the same procedure described in Algorithm 1 but with the following specifications.

- $\mu \in [2, 3.2]$ 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 3 perfectly balanced pairs of

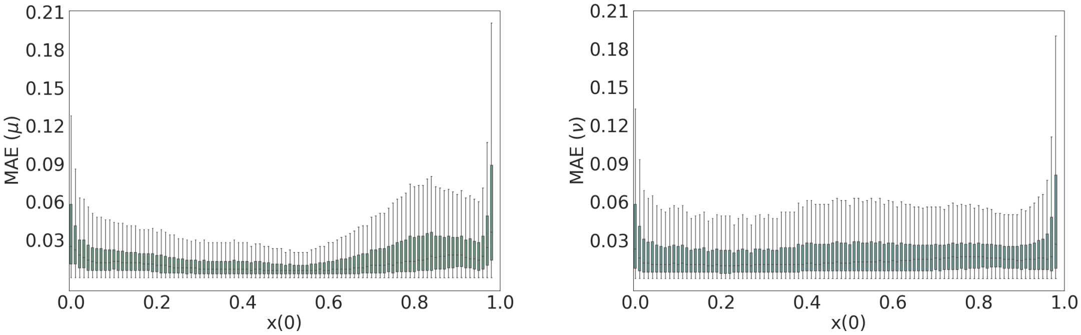


Fig. 8.8: Box and whiskers plots for the MAE distribution obtained from predictions of μ (left) and ν (right) on the evaluation data set in terms of the initial condition $x(0)$. The boxes are painted in green and the whiskers on grey.

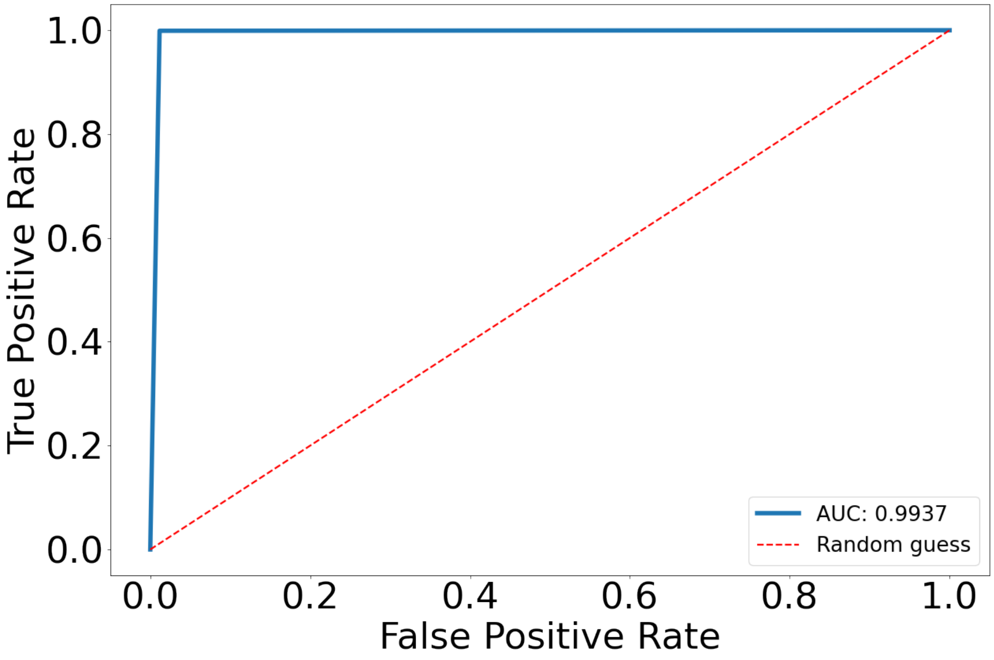


Fig. 8.9: Receiver Operating Characteristic (ROC) curve on the test data set for the classification problem of fractional logistic trajectories with and without delay.

sets, consisting of 618,199 trajectories for each training data set, 142,800 for each validation data set, 190,334 for each the testing data set. For this classification task we have used the architecture shown in Figure 8.3 but changing the activation function in the last dense layer to a sigmoid function. The training finishes after 43 epochs and 3 hours and 30 minutes 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 8.9 where we show the receiver operating characteristic (ROC) curve and the AUC value.

8.4 Conclusions

In this work, we have considered a delayed version of the fractional logistic equation introduced by Wu and Baleanu in [147]. We have seen that a combination of convolutional and recurrent neural networks succeeds in training a model that given an 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 neighbour bins, and this happens almost independently of the values of the parameters μ and ν, see Figure 8.6. 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 MCIN/AEI/ 10.13039/501100011033 and by "ERDF A way of making Europe", by the "European Union".

9 Journal article (v)

Los seres humanos no nacen para siempre el día en que sus madres los alumbran,
sino que la vida los obliga a parirse a sí mismos una y otra vez.

Gabriel García Márquez.

On a new paradigm in the logistic and similar maps: time-stepping schemes

On a new paradigm in the logistic and similar maps: time-stepping schemes

Conejero, J.A.[2], Garibo-i-Orts, O.[1,2], Lizama, C.[3]

1 GRID - Grupo de Investigación en Ciencia de Datos, Valencian International University - VIU, Carrer Pintor Sorolla 21, 46002 València, Spain.
2 Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain.
3 Departamento de Matemática y Ciencia de la Computación Universidad de Santiago de Chile, Las Sophoras 173, Estación Central, Santiago, Chile.

Abstract.
We propose the mathematical analysis of discrete models that consider Lubich's quadrature time-stepping schemes instead of the classical Euler scheme of order 1.
As the first study with this new paradigm, we compare the bifurcation diagrams for the logistic and sine maps obtained from Euler discretizations of orders 1, 2, and 1/2.

We analyze discrete models that consider Lubich's quadrature time-stepping schemes instead of the classical Euler scheme of order 1. We show the existence of chaos for the logistic and sine maps obtained from Euler discretizations of orders 1, 2, and 1/2.

9.1 Introduction

Since Verhulst's introduction of the logistic equation and later popularized by May, a great deal of research has been done from a mathematical and applied point of view. An essential task to understand this model's dynamics is to construct its bifurcation diagram. In fact, given an initial condition y_0, the iterations obtained through

$$y_{n+1} = \eta y_n(1 - y_n), \quad n \in \mathbb{N}_0, \tag{9.1}$$

show regimes of stability, periodicity, and chaos, depending on the values of the control parameter $\eta > 0$. It can be noted that after the change of variable $x_n = \frac{\eta}{\eta-1} y_n$ the equation (9.1) is equivalent to

$$x_{n+1} - x_n = \mu x_n(1 - x_n) \tag{9.2}$$

where $\mu := \eta - 1$, see [9, 134]. Our key observation - and starting point - in this article is that the term $x_{n+1} - x_n$ represents a discretization of the first order derivative, say u', inherited from the continuous model

$$u'(t) = \mu f(u(t)), \quad t \geq 0, \tag{9.3}$$

where f is a given real-valued function. The discretization with step size one used on the left-hand side of (9.2) corresponds to what is commonly known as Euler's method in numerical analysis. From this observation, it is natural to ask: *what happens if we change the Euler discretization method for another?*

In this work, we start with the formula

$$\sum_{j=0}^{n+1} b_{n+1-j} x_j = \mu f(x_n), \quad n \in \mathbb{N}_0, \tag{9.4}$$

where $(b_n)_n$ is a given sequence, and we compare bifurcation diagrams of the associated dynamical systems for some representative cases. It is worth noting that the left-hand side of (9.4) contains an important number of classical time-stepping schemes. For example, if we take $b_n = \delta_{0n} - \delta_{1n}$, where δ_{jn} denotes the Kronecker delta, we get (9.2) for $f(x) := \mu x(1-x)$. From a numerical point of view, the sequences b_n are called quadrature weights. They arise from Lubich's quadrature methods [155], and we can determine them from the following generating power series (with step size $\tau = 1$)

$$G(\xi) = \sum_{n=0}^{\infty} b_n \xi^n, \tag{9.5}$$

named the symbol, or characteristic function, of the scheme [156]. For example for $G(\xi) = 1 - \xi$ we must have $b_n = \delta_{0n} - \delta_{1n}$, described previously. Other examples are the second order difference scheme which is given by the symbol $G(\xi) = (1-\xi) + \frac{1}{2}(1-\xi)^2$ (see [156]) and that produces $b_n = \frac{3}{2}\delta_{0n} - 2\delta_{1n} + \frac{1}{2}\delta_{2n}$; and the Euler scheme of order 1/2 given by the symbol $G(\xi) = (1-\xi)^{1/2}$ that produces $b_n = \frac{\Gamma(n-1/2)}{\Gamma(-1/2)n!}$. The first two cases are named local because they have a finite number of b_n different from zero. The third case is named non-local, and they usually incorporate memory effects in the model. For other examples, we refer to the works of Jin [156], Murillo-Arcila and Lizama [157] about maximal regularity of time-stepping schemes, the works of Wu and Baleanu that introduced a discrete version of the left Caputo differential operator [134, 147], and the work of Nieto for the solution of the logistic differential equation of fractional order without singular kernel [158, 159].

9.2 Methodology

In order to solve (9.4), we assume that there exists a sequence a_n such that $a_n * b_n = \delta_{0n}$ where $a_n * b_n = \sum_{j=0}^{n} a_{n-j} b_j$ denotes the convolution product between a_n and b_n. Then, given x_0 and convolving with a_n in (9.4) we obtain

$$x_n = \mu \sum_{j=1}^{n} a_{n-j} f(x_{j-1}), \quad n \in \mathbb{N}. \tag{9.6}$$

Note that a_n always exists if $b_0 \neq 0$ and is given by the recurrence

$$a_0 = \frac{1}{b_0} \qquad a_n = -\frac{1}{b_0} \sum_{j=1}^{n} a_{n-j} b_j. \tag{9.7}$$

We will compare the bifurcation diagrams of the logistic map $f(x) = \mu x(1-x)$ and sine map $f(x) = \mu \sin(x)$ for the Euler approximation schemes for the first derivative operator of orders $1, 2,$

and 1/2. Using in (9.7) the sequences b_n obtained in the previous section, note that we have $a_n = 1$ in the first case, $a_n = 1 - \frac{1}{3^{n+1}}$ in the second case [160, Example 4.2], and $a_n = \frac{(2n)!}{4^n (n!)^2}$ in the third case [160, Example 4.1]. In this way, we obtain, respectively, the general schemes shown in Table 9.1.

Scheme	General solution
Euler (1st order)	$x_n = \mu \sum_{j=1}^{n} f(x_{n-1})$
Euler (2nd order)	$x_n = \mu \sum_{j=1}^{n} \left(1 - \left(\frac{1}{3}\right)^{n-j+1}\right) f(x_{j-1})$
Euler (1/2 order)	$x_n = \mu \sum_{j=1}^{n} \frac{(2(n-j))!}{4^{(n-j)}\left((n-j)!\right)^2} f(x_{j-1})$

Table 9.1: Euler approximation schemes of orders 1, 2 and 1/2, with the explicit formulas of the general solution.

9.3 Results: Bifurcation diagrams

For constructing the bifurcation diagrams, we consider pairs (x_0, μ), where x_0 is an initial condition and μ is taking in an interval where we want to illustrate the dynamics. For some diagrams presented in this work, we have taken several initial conditions, not only one, to improve the illustration of the dynamics. Given a pair (x_0, μ), we have computed the first 200 terms of each trajectory and represented the last 50 terms, namely $x_{151}, \ldots, x_{200}$. In Figure 9.1, we have the bifurcation diagrams for first-order Euler models, $x_n = \mu \sum_{k=1}^{n} x_{k-1}(1 - x_{k-1})$ and $x_n = \mu \sum_{k=1}^{n} \sin(x_{k-1})$.

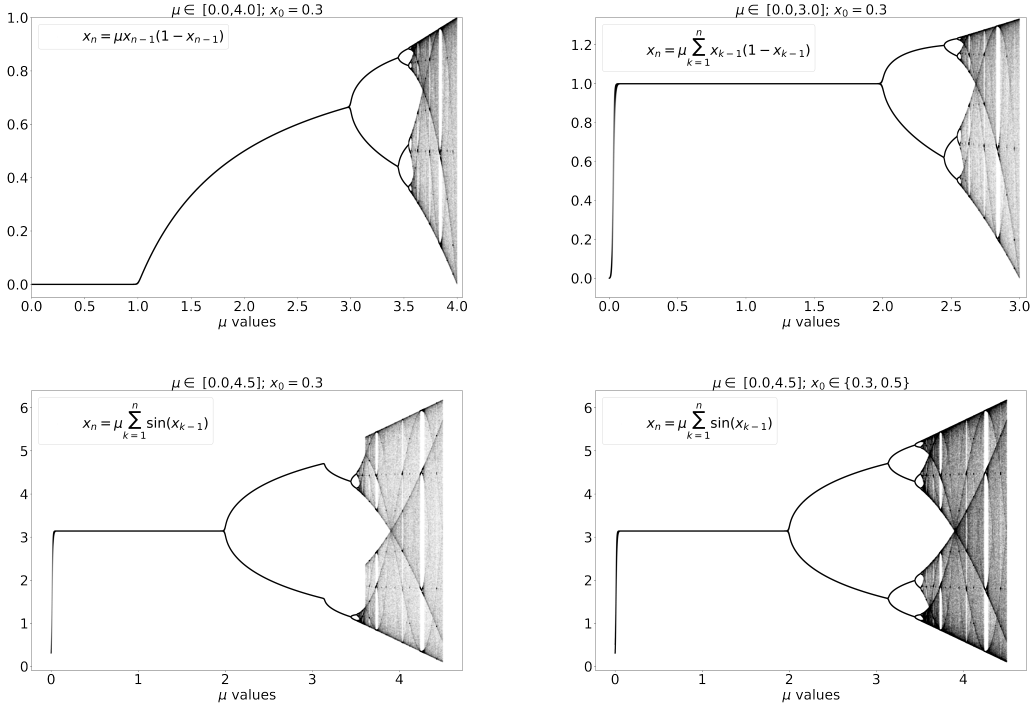


Fig. 9.1: Bifurcation diagrams for the logistic map (top left), the first-order logistic map (top right), and the first-order sine map for $x_0 = 0.3$ (bottom left) and $x_0 \in \{0.3, 0.5\}$ (bottom right). The μ step size is 10^{-4}.

The first is the logistic map diagram; the second can also be found in [146]. We can observe that they are very similar since, as we have commented with the equivalences of (9.1) and (9.2), we can pass from one to the other one with a change of variable.

In order to avoid numerical precision problems in the computation of the bifurcation diagrams of second-order models, we have used the recurrent formula $x_n = \frac{2}{3}\left(2x_{n-1} - \frac{1}{2}x_{n-2} + \mu f(x_{n-1})\right)$ that directly appears from (9.4). Although a second-order model is numerically more stable than a first-order one, we can still find chaos in some regions, as shown in Figure 9.2. We have simultaneously used several initial conditions to get better bifurcation diagram plots. For the case of the logistic term, we can find a chaotic region for $\mu \in [3.83, 3.85]$. Here, the step size for μ has been decreased up to 10^{-5}. For the sinus term, we show chaos in some regions of $\mu \in [6, 7]$ with a μ step size of 10^{-4}. For better plotting the behavior in the interval $[6.18, 6.22]$ we have used 100 initial conditions, $x_0 \in \{0.01, 0.02, \ldots, 0.99\}$, computing the first 200 times of each trajectory, and plotting the last 25 terms of each one.

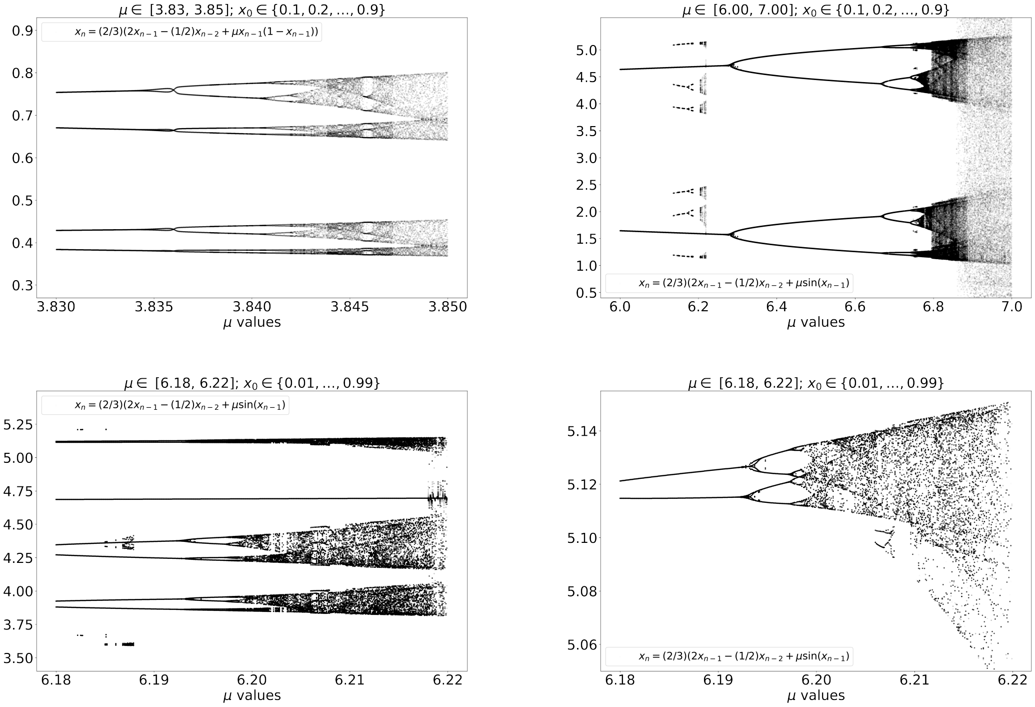


Fig. 9.2: Bifurcation diagrams for the second-order logistic map (top left) and for the second-order sine map (top right). In this second case, we can find smaller regions in the interval $[6.18, 6.22]$ (bottom left and right).

For the half-order case, where we have introduced the memory effect, the model resembles the fractional logistic models inspired by the fractional version of (9.2) [147], showing a blurry part in the region of chaos, as we can see in Figure 9.3.

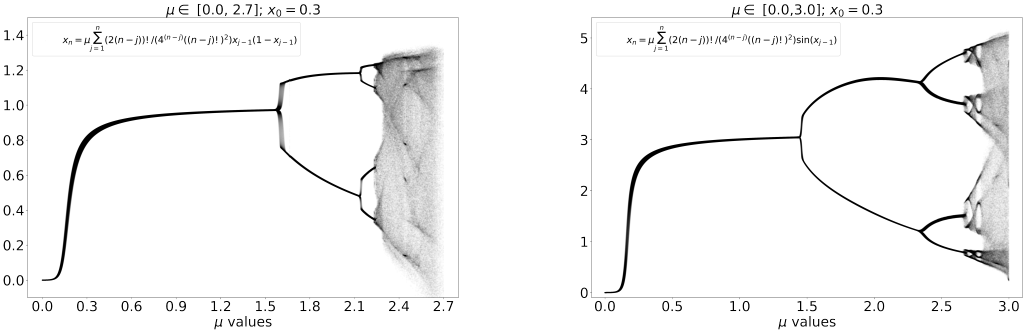


Fig. 9.3: Bifurcation diagrams for the half-order logistic map (left) and for the half-order sine map (right).

9.4 Analysis and conclusions

We have performed a comparative analysis of the dynamics exhibited by some time-stepping schemes originating from quadrature methods. From the recent literature, it is known that these methods are related to some linear viscoelasticity models. For instance, the Euler method is linked with a Newtonian fluid, the second order difference scheme is related to a Maxwell fluid, and the Euler scheme of order 1/2 with a power type material function, see [160]. On the other hand, from [161], it is well known that each one of the above-mentioned models has an associated material function. For instance, for a Newtonian fluid, the material function is $a(t) = 1$, for a Maxwell fluid $a(t) = 1 - e^{-ct}$ where $c \in \mathbb{R}$ is a constant, and for a power type material, the function is $a(t) = \frac{t^\alpha}{\Gamma(\alpha+1)}$ where $\alpha \in (0, 1)$, see [161, Section 5.2].

Now, we observe that each sequence a_n obtained for the time-stepping schemes in (9.6) corresponds in some sense to a discrete version of this material functions, namely, for $a(t) := 1$ we have $a_n = 1$; for $a(t) := 1 - e^{2t/3}$ we have $a_n = 1 - \frac{1}{3^{n+1}}$ and for $a(t) := \frac{t^{1/2}}{\Gamma(3/2)}$ we have $a_n = \frac{(2n)!}{4^n(n!)^2}$. We can even give a mathematical argument to prove this correspondence. For that, we consider the Poisson transformation [162] of a real-valued function $f : [0, \infty) \to \mathbb{R}$ defined by

$$\hat{f}(n) := \int_0^\infty p_n(t) f(t) dt, \quad n \in \mathbb{N},$$

where $p_n(t) = \frac{t^n}{n!} e^{-t}$ and note that a computation produces $\hat{a}(n) = 1 = a_n$, $\hat{a}(n) = 1 - \frac{1}{3^n} = a_{n-1}$ and $\hat{a}(n) = \frac{\Gamma(1/2+n)}{\Gamma(1/2)n!} = \frac{(2n)!}{4^n(n!)^2} = a_n$, respectively. We summarize the characteristic of the models and diagrams in Table 9.2.

Scheme	Viscoelastic Model	Material function	Discrete kernel	Bifurcation diagram
Euler	Newtonian fluid	$a(t) = 1$	$a_n = 1$	Figure 9.1
Second order	Maxwell fluid	$a(t) = 1 - e^{-2t/3}$	$a_n = 1 - \frac{1}{3^{n+1}}$	Figure 9.2
Half order	Power type	$a(t) = \frac{t^{1/2}}{\Gamma(3/2)}$	$a_n = \frac{\Gamma(1/2+n)}{\Gamma(1/2)n!}$	Figure 9.3

Table 9.2: Comparative table of the numerical schemes and properties.

In all cases, we have obtained explicit formulas for computing all the terms of the trajectory, as shown in 9.1. We have also shown the chaos phenomena in all these cases through bifurcation diagrams. In the models from half and first-order schemes, the bifurcation diagrams given by the logistic and sine functions present similar shapes, albeit with different parameter values. However, the similarities disappear when we observe the same maps with the second-order difference schemes. In these last cases, chaos is also present, but one has to look for small regions where it is present. Such dynamical systems show less chaotic behavior and are less powerful for encryption [163]. It will be interesting to study the dynamical properties of these dynamical systems and the connections with the physical properties of the associated viscoelastic models.

Ethics declarations

The authors declare no conflict of interest.

Acknowledgements

JAC acknowledges funding from 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 ANID under FONDECYT grant number 1220036.

10 Concluding remarks and recommendations

Love is the one thing that transcends time and space.
Maybe we should trust that, even if we cannot understand it.

Amelia, Interstellar.

This chapter ends the work carried out in the development of this thesis. A summary of the main concluding remarks is presented below, as well as a set of recommendations to continue the scientific research based on this work.

10.1 Concluding remarks

There is a plethora of problems in the field of physics or mathematics where machine learning can help to shed light. We have focused in problems represented by time series with the aim to target the application of machine learning methods. Anomalous diffusion is a wide spread phenomenon that is present at all scales, from microscopic to macroscopic scale. While statistics based methods were available to characterize the anomalous diffusion, there was a shy appearance of machine learning methods being applied to the problem. The recent availability of new deep learning methods suitable to learn from time series pushed the community to launch an international challenge to set a new state-of-the-art scenario where different methods could be benchmarked. This challenge was the embryo of this thesis. In this thesis we tackle the anomalous diffusion characterization problem from a deep learning perspective, since a deep learning architecture was designed and benchmarked, receiving honors in both the anomalous exponent α regression task and in the classification of the underlying diffusing model task. The defined architecture showed some weakness with short trajectories in both tasks (regression and classification), as the rest of the proposed solutions did. This fact motivated the evaluation of a new approach to the same problem using well-now and widely-used models based on convolutional neural networks which have been used in the field of image processing. Taking advantage of the possibility to represent a time series as an image without loosing temporal information allows the use of these aforementioned deep learning methods. Additionally, using the architecture proposed in this thesis as foundation to develop a model that precisely predicted epidemiological information during the Covid-19 pandemic under the scope of the international XPrize Pandemic Response Challenge showed the robustness of the method. Finally, the extension of the use of the proposed architecture to the logistic map problem, from the field of mathematics, confirms the versatility of the architecture to handle time series and retrieve key paramters that define the behaviour of the logistic map, even at chaotic zones.

This thesis has contributed to the fields of Computer Science, Applied Mathematics and Applied Physics by the innovative application of deep learning technologies. The publication of this work in top-ranked journals and their diffusion in international conferences endorses the community research interest.

The specific concluding remarks of this thesis are listed as follows.

CR1 - Statistical methods to characterize anomalous diffusion motion are being replaced by the use of machine learning methods, mainly deep learning methods. In this work, we have presented our ConvLSTM architecture, a combination of convolutional and recurrent neural networks to accurately characterize anomalous diffusing processes using raw data, with no previous preprocessing or feature extraction. Two key ideas can be highlighted:

- RQ1 and RQ2 were answered with the work presented in chapters 5 and 6. In chapter 6 we presented our architecture ConvLSTM which scored state-of-the-art results during the AnDi Challenge.

  This key idea fulfill research objectives O1, O2 and O3, and derived in the journal publication P1.

- In chapter 6 we introduced the use of Gramian Angular Fields in combination with deep convolutional models which had previously used in the image processing field to improve the performance of the state-of-the-art models for classification of short anomalous diffusing trajectories.

  This key idea fulfill research objective O5, and derived in the journal publication P2.

CR2 - The architecture implemented in this thesis proved to achive an state-of-the-art performance in the characterization of anomalous diffusing processes. But our aim was to extend the use of this architecture to other problems of a different nature. The ConvLSTM architecture was used in the XPrize Pandemic Response Challenge to accurately predict Covid-19 daily cases for 236 regions and countries with minor modifications. Additionally, the same architecture was used to infer the parameters that define the Wu-Baleanu trajectories, related to the logistic map. To the best of our knowledge this has been the first time a deep learning model was used in this problem. Two key ideas can be highlighted:

- RQ3 was answered with the work presented in chapter 4, 7 and 8. In chapter 4 we introduced our participation in the XPrize Pandemic Response Challenge, being awarded as winners.

  This key idea fulfills research objective O4, and derived in two publications which have not been included in this thesis. One of them received honors as the Best Applied Data Science Paper in the ECML-PKDD conference.

- In chapters 7 and 8 the ConvLSTM architecture was used to infer the μ and ν parameters from Wu-Baleanu trajectories, with and without delay. Additionally, the same architecture was used to classify between them achieving optimal performance.

  This key idea fulfills research objective O6, and derived in the journal publications P3, P4 and P5.

10.2 Recommendations

The use of machine learning methods, to be more precise, deep learning methods, can help to shed light to some problem in mathematics and physics. We have proved that a correctly designed architecture can achieve state-of-the-art performance both in regression and classification tasks when handling time series data. Since new deep learning approaches arise continuosly this work can not be considered as an end point, but as an starting point. In this sense, the following lines are suggested for future investigations.

R1 While LSTM recurrent neural networks have proved long term memory and good performance, Transformers are gaining more importance nowadays. In fact, a work that has not been included in this thesis but which I have coauthored presented a transformer used with one dimensional anomalous diffusing trajectories.
In this regard, it would be desirable to work in the extension of the Transformers to two and three dimensional trajectories, which to our knowledge is not yet possible.

R2 We have proved that representing time series by using Gramian Angular Fields, in combination with deep convolutional neural networks architectures achieves outstanding performance with short one dimensional trajectories.

As for the Transformer, it would be interesting to explore the use of this pipeline with two and three dimensional trajectories. Additionally, including Markov Transition Fields images to Gramian Angular Fields could help to improve the yet excelent performance of the method.

R3 Wu-Baleanu trajectories have accurately being characterized by the regression of their μ and ν parameters. In this work we have showed how to infer them for non-delayed and delayed Wu-Baleanu trajectories.

As the continuation of this work one could think of other mathematical problems represented by time series. One could also not try to infer the paremeters but predict the n next terms for a given trajectory.

References

[1] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, “Handwritten digit recognition with a back-propagation network,” *Advances in neural information processing systems*, vol. 2, 1989.

[2] Y. LeCun and Y. Bengio, *Convolutional networks for images, speech, and time series*, p. 255â€“258. Cambridge, MA, USA: MIT Press, 1998.

[3] A. Krizhevsky, V. Nair, and G. Hinton, “Cifar-10 (canadian institute for advanced research),” 2010.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[5] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A critical review of recurrent neural networks for sequence learning,” *arXiv preprint arXiv:1506.00019*, 2015.

[7] J. Brownlee, “Long short-term memory networks with python,” *Machine learning mastery*, 2017.

[8] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, “Lstm: A search space odyssey,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, 2016.

[9] J. A. Conejero, O. Garibo-i Orts, and C. Lizama, “Inferring the fractional nature of wu baleanu trajectories,” *Nonlinear Dyn.*, 2022.

[10] 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, *et al.*, “Objective comparison of methods to decode anomalous diffusion,” *Nat. Commun.*, vol. 12, no. 1, p. 6253, 2021.

[11] R. B. Davies and D. Harte, “Tests for Hurst effect,” *Biometrika*, vol. 74, no. 1, pp. 95–101, 1987.

[12] J. R. Hosking, “Modeling persistence in hydrological time series using fractional differencing,” *Water Resour. Res.*, vol. 20, no. 12, pp. 1898–1908, 1984.

[13] H. Verdier, M. Duval, F. Laurent, A. Cassé, C. L. Vestergaard, and J.-B. Masson, “Learning physical properties of anomalous random walks using graph neural networks,” *J. Phys. A-Math. Theor.*, vol. 54, no. 23, p. 234001, 2021.

[14] A. Argun, G. Volpe, and S. Bo, “Classification, inference and segmentation of anomalous diffusion with recurrent neural networks,” *J. Phys. A-Math. Theor.*, vol. 54, no. 29, p. 294003, 2021.

[15] C. Manzo, “Extreme learning machine for the characterization of anomalous diffusion from single trajectories (AnDi-ELM),” *J. Phys. A-Math. Theor.*, vol. 54, no. 33, p. 334002, 2021.

[16] A. Gentili and G. Volpe, “Characterization of anomalous diffusion classical statistics powered by deep learning (CONDOR),” *J. Phys. A-Math. Theor.*, vol. 54, no. 31, p. 314003, 2021.

[17] 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.*, vol. 54, no. 50, p. 504002, 2021.

[18] P. Kowalek, H. Loch-Olszewska, L. Laszczuk, J. Opala, and J. Szwabiński, “Boosting the performance of anomalous diffusion classifiers with the proper choice of features,” *J. Phys. A-Math. Theor.*, vol. 55, p. 244005, may 2022.

[19] N. Firbas, Ò. Garibo i Orts, M. A. Garcia-March, and J. A. Conejero, "Characterization of anomalous diffusion through convolutional transformers," *J. Phys. A-Math. Theor.*, 2022.

[20] M. Lozano, O. Garibo-i Orts, E. Piñol, M. Rebollo, K. Polotskaya, M. Garcia-March, J. Conejero, F. Escolano, and 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*, pp. 384–399, 2021.

[21] C. Sáez, N. Romero, J. A. Conejero, and J. M. García-Gómez, "Potential limitations in COVID-19 machine learning due to data source variability: A case study in the nCov2019 dataset," *J. Am. Med. Inf. Assoc.*, vol. 28, no. 2, pp. 360–364, 2021.

[22] L. Zhou, N. Romero-García, J. Martínez-Miranda, J. A. Conejero, J. M. García-Gómez, C. Sáez, *et al.*, "Subphenotyping of Mexican patients with COVID-19 at preadmission to anticipate severity stratification: Age-sex unbiased meta-clustering technique," *JMIR Public Health Surveill.*, vol. 8, no. 3, p. e30032, 2022.

[23] D. Hervás, M. Rebollo, J. A. Conejero, N. Oliver, *et al.*, "COVID-19 outbreaks analysis in the Valencian Region of Spain in the prelude of the third wave," *Front. Public Health*, p. 4503, 2022.

[24] N. Oliver, X. Barber, K. Roomp, K. Roomp, *et al.*, "Assessing the impact of the COVID-19 pandemic in Spain: large-scale, online, self-reported population survey," *J. Med. Internet Res.*, vol. 22, no. 9, p. e21319, 2020.

[25] M. Martinez-Garcia, A. Rabasa, X. Barber, K. Polotskaya, K. Roomp, and N. Oliver, "Key factors affecting peopleâ€™s unwillingness to be confined during the COVID-19 pandemic in Spain: A large-scale population study," *Sci Rep*, vol. 11, no. 1, p. 18626, 2021.

[26] M. V. Ibañez, M. Martínez-Garcia, and A. Simó, "A review of spatiotemporal models for count data in R packages. a case study of COVID-19 data," *Mathematics*, vol. 9, no. 13, p. 1538, 2021.

[27] R. Miikkulainen, O. Francon, E. Meyerson, X. Qiu, D. Sargent, E. Canzani, and B. Hodjat, "From prediction to prescription: evolutionary optimization of nonpharmaceutical interventions in the COVID-19 pandemic," *IEEE Trans. Evol. Comput.*, vol. 25, no. 2, pp. 386–401, 2021.

[28] S. A. Lauer, K. H. Grantz, Q. Bi, F. K. Jones, Q. Zheng, H. R. Meredith, A. S. Azman, N. G. Reich, and J. Lessler, "The incubation period of coronavirus disease 2019 (COVID-19) from publicly reported confirmed cases: estimation and application," *Ann Intern Med.*, vol. 172, no. 9, pp. 577–582, 2020.

[29] M. Islam, "Einstein–Smoluchowski diffusion equation: a discussion," *Phys. Scr.*, vol. 70, no. 2-3, p. 120, 2004.

[30] M. Islam, "Fickian diffusion equationâ€"an unsolved problem," *Phys. Scr.*, vol. 70, no. 2-3, p. 114, 2004.

[31] H. Scher and E. W. Montroll, "Anomalous transit-time dispersion in amorphous solids," *Phys. Rev. B*, vol. 12, no. 6, p. 2455, 1975.

[32] J. Klafter and G. Zumofen, "Lévy statistics in a hamiltonian system," *Phys. Rev. E*, vol. 49, no. 6, p. 4873, 1994.

[33] P. Massignan, C. Manzo, J. A. Torreno-Pina, M. F. García-Parajo, M. Lewenstein, and G. Lapeyre Jr, "Nonergodic subdiffusion from Brownian motion in an inhomogeneous medium," *Phys. Rev. Lett.*, vol. 112, no. 15, p. 150603, 2014.

[34] G. Muñoz-Gil, M. A. Garcia-March, C. Manzo, A. Celi, and M. Lewenstein, "Diffusion through a network of compartments separated by partially-transmitting boundaries," *Front. Physics*, vol. 7, p. 31, 2019.

[35] C. Charalambous, G. Muñoz-Gil, A. Celi, M. Garcia-Parajo, M. Lewenstein, C. Manzo, and M. García-March, "Nonergodic subdiffusion from transient interactions with heterogeneous partners," *Phys. Rev. E*, vol. 95, no. 3, p. 032403, 2017.

[36] G. Muñoz-Gil, C. Charalambous, M. García-March, M. Garcia-Parajo, C. Manzo, M. Lewenstein, and A. Celi, "Transient subdiffusion from an Ising environment," *Phys. Rev. E*, vol. 96, no. 5, p. 052140, 2017.

[37] B. B. Mandelbrot and J. W. Van Ness, “Fractional Brownian motions, fractional noises and applications,” *SIAM Rev.*, vol. 10, no. 4, pp. 422–437, 1968.

[38] J.-H. Jeon and R. Metzler, “Fractional Brownian motion and motion governed by the fractional Langevin equation in confined geometries,” *Phys. Rev. E*, vol. 81, no. 2, p. 021103, 2010.

[39] S. C. Lim and S. V. Muniandy, “Self-similar gaussian processes for modeling anomalous diffusion,” *Phys. Rev. E*, vol. 66, no. 2, p. 021114, 2002.

[40] R. Metzler, J.-H. Jeon, A. G. Cherstvy, and E. Barkai, “Anomalous diffusion models and their properties: non-stationarity, non-ergodicity, and ageing at the centenary of single particle tracking,” *Phys. Chem. Chem. Phys.*, vol. 16, no. 44, pp. 24128–24164, 2014.

[41] E. Kepten, A. Weron, G. Sikora, K. Burnecki, and Y. Garini, “Guidelines for the fitting of anomalous diffusion mean square displacement graphs from single particle tracking experiments,” *PLoS One*, vol. 10, no. 2, p. e0117722, 2015.

[42] V. Tejedor, O. Bénichou, R. Voituriez, R. Jungmann, F. Simmel, C. Selhuber-Unkel, L. B. Oddershede, and R. Metzler, “Quantitative analysis of single particle trajectories: mean maximal excursion method,” *Biophys J.*, vol. 98, no. 7, pp. 1364–1372, 2010.

[43] N. Makarava, S. Benmehdi, and M. Holschneider, “Bayesian estimation of self-similarity exponent,” *Phys. Rev. E*, vol. 84, no. 2, p. 021109, 2011.

[44] K. Hinsen and G. R. Kneller, “Communication: A multiscale bayesian inference approach to analyzing subdiffusion in particle trajectories,” *J Chem Phys.*, vol. 145, no. 15, p. 151101, 2016.

[45] K. Burnecki, E. Kepten, Y. Garini, G. Sikora, and A. Weron, “Estimating the anomalous diffusion exponent for single particle tracking data with measurement errors - An alternative approach,” *Sci. Rep.*, vol. 5, no. 1, pp. 1–11, 2015.

[46] S. Thapa, A. Wyłomańska, G. Sikora, C. E. Wagner, D. Krapf, H. Kantz, A. V. Chechkin, and R. Metzler, “Leveraging large-deviation statistics to decipher the stochastic properties of measured trajectories,” *New J. Phys.*, vol. 23, no. 1, p. 013008, 2021.

[47] J. Gajda, A. Wyłomańska, H. Kantz, A. Chechkin, and G. Sikora, “Large deviations of time-averaged statistics for Gaussian processes,” *Stat. Probab. Lett.*, vol. 143, pp. 47–55, 2018.

[48] E. Agliari, P. J. Sáez, A. Barra, M. Piel, P. Vargas, and M. Castellana, “A statistical inference approach to reconstruct intercellular interactions in cell migration experiments,” *Sci. Adv.*, vol. 6, no. 11, p. eaay2103, 2020.

[49] D. Krapf, E. Marinari, R. Metzler, G. Oshanin, X. Xu, and A. Squarcini, “Power spectral density of a single Brownian trajectory: what one can and cannot learn from it,” *New J. Phys.*, vol. 20, no. 2, p. 023029, 2018.

[50] D. Krapf, N. Lukat, E. Marinari, R. Metzler, G. Oshanin, C. Selhuber-Unkel, A. Squarcini, L. Stadler, M. Weiss, and X. Xu, “Spectral content of a single non-Brownian trajectory,” *Phys. Rev. X*, vol. 9, no. 1, p. 011019, 2019.

[51] E. Kepten, I. Bronshtein, and Y. Garini, “Improved estimation of anomalous diffusion exponents in single-particle tracking experiments,” *Phys. Rev. E*, vol. 87, no. 5, p. 052713, 2013.

[52] A. Weron, J. Janczura, E. Boryczka, T. Sungkaworn, and D. Calebiro, “Statistical testing approach for fractional anomalous diffusion classification,” *Phys. Rev. E*, vol. 99, no. 4, p. 042149, 2019.

[53] Y. Meroz and I. M. Sokolov, “A toolbox for determining subdiffusive mechanisms,” *Phys. Rep.*, vol. 573, pp. 1–29, 2015.

[54] N. Monnier, S.-M. Guo, M. Mori, J. He, P. Lénárt, and M. Bathe, “Bayesian approach to MSD-based analysis of particle motion in live cells,” *Biophys. J.*, vol. 103, no. 3, pp. 616–626, 2012.

[55] T. Bickel, “A note on confined diffusion,” *Physica A*, vol. 377, no. 1, pp. 24–32, 2007.

[56] R. D. Astumian and P. Hänggi, “Brownian motors,” *Phys. Today*, vol. 55, no. 11, pp. 33–39, 2002.

[57] M. Magdziarz, A. Weron, K. Burnecki, and J. Klafter, “Fractional Brownian motion versus the continuous-time random walk: A simple test for subdiffusive dynamics,” *Phys. Rev. Lett.*, vol. 103, no. 18, p. 180602, 2009.

[58] J.-H. Jeon and R. Metzler, “Analysis of short subdiffusive time series: scatter of the time-averaged mean-squared displacement,” *J. Phys. A-Math. Theor.*, vol. 43, no. 25, p. 252001, 2010.

[59] K. Burnecki, E. Kepten, J. Janczura, I. Bronshtein, Y. Garini, and A. Weron, “Universal algorithm for identification of fractional Brownian motion. A case of telomere subdiffusion,” *Biophys. J.*, vol. 103, no. 9, pp. 1839–1847, 2012.

[60] S. Thapa, M. A. Lomholt, J. Krog, A. G. Cherstvy, and R. Metzler, “Bayesian analysis of single-particle tracking data using the nested-sampling algorithm: maximum-likelihood model selection applied to stochastic-diffusivity data,” *Phys. Chem. Chem. Phys.*, vol. 20, no. 46, pp. 29018–29037, 2018.

[61] A. G. Cherstvy, S. Thapa, C. E. Wagner, and R. Metzler, “Non-Gaussian, non-ergodic, and non-Fickian diffusion of tracers in mucin hydrogels,” *Soft Matter*, vol. 15, no. 12, pp. 2526–2551, 2019.

[62] Y. Meroz, I. M. Sokolov, and J. Klafter, “Test for determining a subdiffusive model in ergodic systems from single trajectories,” *Phys. Rev. Lett.*, vol. 110, no. 9, p. 090601, 2013.

[63] L. Chen, K. E. Bassler, J. L. McCauley, and G. H. Gunaratne, “Anomalous scaling of stochastic processes and the Moses effect,” *Phys. Rev. E*, vol. 95, no. 4, p. 042141, 2017.

[64] T. Wagner, A. Kroll, M. Wiemann, and H.-G. Lipinski, “Classification of nanoparticle diffusion processes in vital cells by a multifeature random forests approach: application to simulated data, darkfield, and confocal laser scanning microscopy,” in *Biophotonics: Photonic Solutions for Better Health Care V*, vol. 9887, pp. 190–198, SPIE, 2016.

[65] T. Wagner, A. Kroll, C. R. Haramagatti, H.-G. Lipinski, and M. Wiemann, “Classification and segmentation of nanoparticle diffusion trajectories in cellular micro environments,” *PloS one*, vol. 12, no. 1, p. e0170165, 2017.

[66] G. Muñoz-Gil, M. A. Garcia-March, C. Manzo, J. D. Martín-Guerrero, and M. Lewenstein, “Single trajectory characterization via machine learning,” *New J. Phys.*, vol. 22, no. 1, p. 013010, 2020.

[67] J. Janczura, P. Kowalek, H. Loch-Olszewska, J. Szwabiński, and A. Weron, “Classification of particle trajectories in living cells: Machine learning versus statistical testing hypothesis for fractional anomalous diffusion,” *Phys. Rev. E*, vol. 102, no. 3, p. 032402, 2020.

[68] H. Loch-Olszewska and J. Szwabiński, “Impact of feature choice on machine learning classification of fractional anomalous diffusion,” *Entropy*, vol. 22, no. 12, p. 1436, 2020.

[69] S. Bo, F. Schmidt, R. Eichhorn, and G. Volpe, “Measurement of anomalous diffusion using recurrent neural networks,” *Phys. Rev. E*, vol. 100, no. 1, p. 010102, 2019.

[70] D. Han, N. Korabel, R. Chen, M. Johnston, A. Gavrilova, V. J. Allan, S. Fedotov, and T. A. Waigh, “Deciphering anomalous heterogeneous intracellular transport with neural networks,” *eLife*, vol. 9, p. e52224, 2020.

[71] N. Granik, L. E. Weiss, E. Nehme, M. Levin, M. Chein, E. Perlson, Y. Roichman, and Y. Shechtman, “Single-particle diffusion characterization by deep learning,” *Biophys. J.*, vol. 117, no. 2, pp. 185–192, 2019.

[72] P. Kowalek, H. Loch-Olszewska, and J. Szwabiński, “Classification of diffusion modes in single-particle tracking data: Feature-based versus deep-learning approach,” *Phys. Rev. E*, vol. 100, no. 3, p. 032410, 2019.

[73] F. Cichos, K. Gustavsson, B. Mehlig, and G. Volpe, “Machine learning for active matter,” *Nat. Mach. Intell.*, vol. 2, no. 2, pp. 94–103, 2020.

[74] G. Muñoz-Gil, G. Volpe, M. A. García-March, R. Metzler, M. Lewenstein, and C. Manzo, “The anomalous diffusion challenge: single trajectory characterisation as a competition,” in *Emerging Topics in Artificial Intelligence 2020*, vol. 11469, pp. 42–51, SPIE, 2020.

[75] F. A. Oliveira, R. M. Ferreira, L. C. Lapas, and M. H. Vainstein, "Anomalous diffusion: A basic mechanism for the evolution of inhomogeneous systems," *Front. Physics*, vol. 7, p. 18, 2019.

[76] J. Klafter and I. M. Sokolov, "Anomalous diffusion spreads its wings," *Phys. World*, vol. 18, no. 8, p. 29, 2005.

[77] Y. Sagi, M. Brook, I. Almog, and N. Davidson, "Observation of anomalous diffusion and fractional self-similarity in one dimension," *Phys. Rev. Lett.*, vol. 108, no. 9, p. 093002, 2012.

[78] A. Dechant, F. Kindermann, A. Widera, and E. Lutz, "Continuous-time random walk for a particle in a periodic potential," *Phys. Rev. Lett.*, vol. 123, no. 7, p. 070602, 2019.

[79] F. Kindermann, A. Dechant, M. Hohmann, T. Lausch, D. Mayer, F. Schmidt, E. Lutz, and A. Widera, "Nonergodic diffusion of single atoms in a periodic potential," *Nat. Phys.*, vol. 13, no. 2, pp. 137–141, 2017.

[80] Y. Tang, S. Bao, and W. Guo, "Superdiffusion of quantized vortices uncovering scaling laws in quantum turbulence," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 118, no. 6, p. e2021957118, 2021.

[81] E. Lenzi, P. Fernandes, T. Petrucci, H. Mukai, H. Ribeiro, M. Lenzi, and G. Gonçalves, "Anomalous diffusion and electrical response of ionic solutions," *Int. J. Electrochem. Sci*, vol. 8, pp. 2849–2862, 2013.

[82] I. Bronstein, Y. Israel, E. Kepten, S. Mai, Y. Shav-Tal, E. Barkai, and Y. Garini, "Transient anomalous diffusion of telomeres in the nucleus of mammalian cells," *Phys. Rev. Lett.*, vol. 103, no. 1, p. 018102, 2009.

[83] L. Stadler and M. Weiss, "Non-equilibrium forces drive the anomalous diffusion of telomeres in the nucleus of mammalian cells," *New J. Phys.*, vol. 19, no. 11, p. 113048, 2017.

[84] C. Manzo and M. F. Garcia-Parajo, "A review of progress in single particle tracking: from methods to biophysical insights," *Rep. Prog. Phys.*, vol. 78, no. 12, p. 124601, 2015.

[85] A. V. Weigel, B. Simon, M. M. Tamkun, and D. Krapf, "Ergodic and nonergodic processes coexist in the plasma membrane as observed by single-molecule tracking," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 108, no. 16, pp. 6438–6443, 2011.

[86] C. Manzo, J. A. Torreno-Pina, P. Massignan, G. J. Lapeyre Jr, M. Lewenstein, and M. F. G. Parajo, "Weak ergodicity breaking of receptor motion in living cells stemming from random diffusivity," *Phys. Rev. X*, vol. 5, no. 1, p. 011021, 2015.

[87] A. Caspi, R. Granek, and M. Elbaum, "Enhanced diffusion in active intracellular transport," *Phys. Rev. Lett.*, vol. 85, no. 26, p. 5655, 2000.

[88] S. C. Weber, A. J. Spakowitz, and J. A. Theriot, "Bacterial chromosomal loci move subdiffusively through a viscoelastic cytoplasm," *Phys. Rev. Lett.*, vol. 104, no. 23, p. 238102, 2010.

[89] B. M. Regner, D. Vučinić, C. Domnisoru, T. M. Bartol, M. W. Hetzer, D. M. Tartakovsky, and T. J. Sejnowski, "Anomalous diffusion of single particles in cytoplasm," *Biophys J.*, vol. 104, no. 8, pp. 1652–1660, 2013.

[90] A. Sabri, X. Xu, D. Krapf, and M. Weiss, "Elucidating the origin of heterogeneous anomalous diffusion in the cytoplasm of mammalian cells," *Phys. Rev. Lett.*, vol. 125, no. 5, p. 058101, 2020.

[91] J.-H. Jeon, N. Leijnse, L. B. Oddershede, and R. Metzler, "Anomalous diffusion and power-law relaxation of the time averaged mean squared displacement in worm-like micellar solutions," *New J. Phys.*, vol. 15, no. 4, p. 045011, 2013.

[92] I. M. Tolić-Nørrelykke, E.-L. Munteanu, G. Thon, L. Oddershede, and K. Berg-Sørensen, "Anomalous diffusion in living yeast cells," *Phys. Rev. Lett.*, vol. 93, no. 7, p. 078102, 2004.

[93] M. Köpf, C. Corinth, O. Haferkamp, and T. Nonnenmacher, "Anomalous diffusion of water in biological tissues," *Biophys J.*, vol. 70, no. 6, pp. 2950–2958, 1996.

[94] E. Özarslan, P. J. Basser, T. M. Shepherd, P. E. Thelwall, B. C. Vemuri, and S. J. Blackband, "Observation of anomalous diffusion in excised tissue by characterizing the diffusion-time dependence of the MR signal," *J. Magn. Reson.*, vol. 183, no. 2, pp. 315–323, 2006.

[95] R. L. Magin, C. Ingo, L. Colon-Perez, W. Triplett, and T. H. Mareci, “Characterization of anomalous diffusion in porous biological tissues using fractional order derivatives and entropy,” *Microporous Mesoporous Mat.*, vol. 178, pp. 39–43, 2013.

[96] Z. Zhang and U. Angst, “A dual-permeability approach to study anomalous moisture transport properties of cement-based materials,” *Transp. Porous Media*, vol. 135, no. 1, pp. 59–78, 2020.

[97] V. Méndez, D. Campos, and F. Bartumeus, *Stochastic foundations in movement ecology.* Springer, 2016.

[98] I. Golding and E. C. Cox, “Physical nature of bacterial cytoplasm,” *Phys. Rev. Lett.*, vol. 96, no. 9, p. 098102, 2006.

[99] Y. He, S. Burov, R. Metzler, and E. Barkai, “Random time-scale invariant diffusion and transport coefficients,” *Phys. Rev. Lett.*, vol. 101, no. 5, p. 058101, 2008.

[100] M. Magdziarz and A. Weron, “Anomalous diffusion: testing ergodicity breaking in experimental data,” *Phys. Rev. E*, vol. 84, no. 5, p. 051138, 2011.

[101] D. Molina-García, T. M. Pham, P. Paradisi, C. Manzo, and G. Pagnini, “Fractional kinetics emerging from ergodicity breaking in random media,” *Phys. Rev. E*, vol. 94, no. 5, p. 052147, 2016.

[102] Y. Lanoiselée, N. Moutal, and D. S. Grebenkov, “Diffusion-limited reactions in dynamic heterogeneous media,” *Nat. Commun.*, vol. 9, no. 1, p. 4398, 2018.

[103] A. Godec and R. Metzler, “Finite-time effects and ultraweak ergodicity breaking in superdiffusive dynamics,” *Phys. Rev. Lett.*, vol. 110, no. 2, p. 020603, 2013.

[104] A. Godec and R. Metzler, “Linear response, fluctuation-dissipation, and finite-system-size effects in superdiffusion,” *Phys. Rev. E*, vol. 88, no. 1, p. 012116, 2013.

[105] G. Bel and E. Barkai, “Weak ergodicity breaking in the continuous-time random walk,” *Phys. Rev. Lett.*, vol. 94, no. 24, p. 240602, 2005.

[106] W. Deng and E. Barkai, “Ergodic properties of fractional Brownian-Langevin motion,” *Phys. Rev. E*, vol. 79, no. 1, p. 011112, 2009.

[107] M. Schwarzl, A. Godec, and R. Metzler, “Quantifying non-ergodicity of anomalous diffusion with higher order moments,” *Sci. Rep.*, vol. 7, no. 1, p. 3878, 2017.

[108] Y. Mardoukhi, A. Chechkin, and R. Metzler, “Spurious ergodicity breaking in normal and fractional ornstein–uhlenbeck process,” *New J. Phys.*, vol. 22, no. 7, p. 073012, 2020.

[109] N. Chenouard, I. Smal, F. De Chaumont, M. Maška, I. F. Sbalzarini, Y. Gong, J. Cardinale, C. Carthel, S. Coraluppi, M. Winter, *et al.*, “Objective comparison of particle tracking methods,” *Nat. Methods*, vol. 11, no. 3, pp. 281–289, 2014.

[110] J.-H. Jeon, E. Barkai, and R. Metzler, “Noisy continuous time random walks,” *J. Chem. Phys.*, vol. 139, no. 12, p. 09B616_1, 2013.

[111] G. Muñoz-Gil, C. Romero-Aristizabal, N. Mateos, L. I. de Llobet Cucalon, M. Beato, M. Lewenstein, M. F. Garcia-Parajo, and J. A. Torreno-Pina, “Phase separation of tunable biomolecular condensates predicted by an interacting particle model,” *bioRxiv*, pp. 2020–09, 2020.

[112] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[113] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, *et al.*, “Huggingface’s transformers: State-of-the-art natural language processing,” *arXiv preprint arXiv:1910.03771*, 2019.

[114] R. Brown, “Xxvii. a brief account of microscopical observations made in the months of June, July and August 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies,” *Philosoph. Mag.*, vol. 4, no. 21, pp. 161–173, 1828.

[115] E. G. Flekkøy, A. Hansen, and B. Baldelli, “Hyperballistic superdiffusion and explosive solutions to the non-linear diffusion equation,” *Front. Phys.*, vol. 9, p. 640560, 2021.

[116] O. Vilk, Y. Orchan, M. Charter, N. Ganot, S. Toledo, R. Nathan, and M. Assaf, “Ergodicity breaking in area-restricted search of avian predators,” *Phys. Rev. X*, vol. 12, p. 031005, Jul 2022.

[117] 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,” in *Emerging Topics in Artificial Intelligence (ETAI) 2021*, vol. 11804, p. 1180416, SPIE, 2021.

[118] O. Vilk, E. Aghion, T. Avgar, C. Beta, O. Nagel, A. Sabri, R. Sarfati, D. K. Schwartz, M. Weiss, D. Krapf, *et al.*, “Unravelling the origins of anomalous diffusion: from molecules to migrating storks,” *Phys. Rev. Res.*, vol. 4, no. 3, p. 033055, 2022.

[119] F. Thiel and I. M. Sokolov, “Weak ergodicity breaking in an anomalous diffusion process of mixed origins,” *Phys. Rev. E*, vol. 89, p. 012136, Jan 2014.

[120] S. Benmehdi, N. Makarava, N. Benhamidouche, and M. Holschneider, “Bayesian estimation of the self-similarity exponent of the Nile River fluctuation,” *Nonlinear Process. Geophys.*, vol. 18, no. 3, pp. 441–446, 2011.

[121] Z. Wang and T. Oates, “Imaging time-series to improve classification and imputation,” in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[122] Y.-Y. Hong, J. J. F. Martinez, and A. C. Fajardo, “Day-ahead solar irradiation forecasting utilizing gramian angular field and convolutional long short-term memory,” *IEEE access*, vol. 8, pp. 18741–18753, 2020.

[123] G. Zhang, Y. Si, D. Wang, W. Yang, and Y. Sun, “Automated detection of myocardial infarction using a gramian angular field and principal component analysis network,” *IEEE Access*, vol. 7, pp. 171570–171583, 2019.

[124] K. P. Thanaraj, B. Parvathavarthini, U. J. Tanik, V. Rajinikanth, S. Kadry, and K. Kamalanand, “Implementation of deep neural networks to classify EEG signals using gramian angular summation field for epilepsy diagnosis,” *arXiv preprint arXiv:2003.04534*, 2020.

[125] H. Xu, J. Li, H. Yuan, Q. Liu, S. Fan, T. Li, and X. Sun, “Human activity recognition based on gramian angular field and deep convolutional neural network,” *IEEE Access*, vol. 8, pp. 199393–199405, 2020.

[126] S. D. Wickramaratne and M. S. Mahmud, “A deep learning based ternary task classification system using gramian angular summation field in fNIRS neuroimaging data,” in *2020 IEEE International Conference on E-health Networking, Application & Services (HEALTHCOM)*, pp. 1–4, 2021.

[127] S. Liu, S. Wang, C. Hu, and W. Bi, “Determination of alcohols-diesel oil by near infrared spectroscopy based on gramian angular field image coding and deep learning,” *Fuel*, vol. 309, p. 122121, 2022.

[128] F. Peng, Z. Zhang, X. Gou, H. Liu, and W. Wang, “Motion artifact removal from photoplethysmographic signals by combining temporally constrained independent component analysis and adaptive filter,” *Biomed. Eng. Online*, vol. 13, no. 1, pp. 1–14, 2014.

[129] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[130] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[131] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint arXiv:1803.01271*, 2018.

[132] R. M. May, “Simple mathematical models with very complicated dynamics,” *Nature*, vol. 261, pp. 459–467, 1976.

[133] S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering.* CRC press, 2018.

[134] G.-C. Wu and D. Baleanu, “Discrete fractional logistic map and its chaos,” *Nonlinear Dynam.*, vol. 75, pp. 283–287, 2014.

[135] J. A. Conejero, C. Lizama, A. Mira-Iglesias, and C. Rodero, “Visibility graphs of fractional Wu–Baleanu time series,” *J. Differ. Equ. Appl.*, vol. 25, no. 9-10, pp. 1321–1331, 2019.

[136] O. Garibo-i Orts, N. Firbas, L. Sebasti´ a, and J. A. Conejero, “Gramian Angular Fields for leveraging pre-trained computer vision models with anomalous diffusion trajectories,” 2022.

[137] 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*, vol. 133, no. 9, pp. 1–21, 2018.

[138] Z. Sabir, M. Raja, D. Baleanu, K. Cengiz, and M. Shoaib, “Design of Gudermannian Neuroswarming to solve the singular Emden–Fowler nonlinear model numerically,” *Nonlinear Dynam.*, vol. 106, no. 4, pp. 3199–3214, 2021.

[139] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016.

[140] A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F. Ruiz, J. Schrittwieser, G. Swirszcz, *et al.*, “Discovering faster matrix multiplication algorithms with reinforcement learning,” *Nature*, vol. 610, no. 7930, pp. 47–53, 2022.

[141] B. Kuttner, “On differences of fractional order,” *Proc. London Math. Soc.*, vol. 3, no. 1, pp. 453–466, 1957.

[142] J. B. Díaz and T. Osler, “Differences of fractional order,” *Math. Comp.*, vol. 28, no. 125, pp. 185–202, 1974.

[143] H. L. Gray and N. F. Zhang, “On a new definition of the fractional difference,” *Math. Comp.*, vol. 50, no. 182, pp. 513–529, 1988.

[144] K. S. Miller and B. Ross, “Fractional difference calculus,” in *Proceedings of the international symposium on univalent functions, fractional calculus and their applications*, pp. 139–152, 1988.

[145] A. F. M and P. W. Eloe, “A transform method in discrete fractional calculus,” *Int. J. Difference Equ.*, vol. 2, no. 2, 2007.

[146] G.-C. Wu, D. Baleanu, and S.-D. Zeng, “Discrete chaos in fractional sine and standard maps,” *Phys. Lett. A*, vol. 378, no. 5-6, pp. 484–487, 2014.

[147] G.-C. Wu and D. Baleanu, “Discrete chaos in fractional delayed logistic maps,” *Nonlinear Dynam.*, vol. 80, pp. 1697–1703, 2015.

[148] A. Atangana, *Derivative with a new parameter: theory, methods and applications.* Academic Press, 2015.

[149] D. Baleanu, Z. B. Güvenç, J. T. Machado, *et al.*, *New trends in nanotechnology and fractional calculus applications*, vol. 10. Springer, 2010.

[150] J. Conejero, J. Franceschi, and E. Picó-Marco, “Fractional vs. ordinary control systems: What does the fractional derivative provide?,” *Mathematics*, vol. 10, no. 15, p. 2719, 2022.

[151] E. İlhan and İ. Kıymaz, “A generalization of truncated M-fractional derivative and applications to fractional differential equations,” *Applied Math. Nonlinear Sci.*, vol. 5, no. 1, pp. 171–188, 2020.

[152] 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.*, vol. 64, pp. 213–231, 2018.

[153] D. Valério, M. Ortigueira, and A. Lopes, “How many fractional derivatives are there?,” *Mathematics*, vol. 10, no. 5, p. 737, 2022.

[154] M. Ortigueira and J. Machado, “Which derivative?,” *Fract. Fractional*, vol. 1, no. 1, p. 3, 2017.

[155] C. Lubich, “Convolution quadrature and discretized operational calculus. I,” *Numer. Math. (Heidelb.)*, vol. 52, no. 2, pp. 129–145, 1988.

[156] B. Jin, B. Li, and Z. Zhou, “Discrete maximal regularity of time-stepping schemes for fractional evolution equations,” *Numer. Math. (Heidelb.)*, vol. 138, pp. 101–131, 2018.

[157] C. Lizama and M. Murillo-Arcila, “Maximal regularity for time-stepping schemes arising from convolution quadrature of non-local in time equations,” *Discrete Contin. Dyn. Syst. Ser. A*, vol. 42, no. 8, pp. 3787–3807, 2022.

[158] J. Nieto, “Solution of a fractional logistic ordinary differential equation,” *Appl. Math. Let.*, vol. 123, p. 107568, 2022.

[159] T. Zhang and Y. Li, “Exponential Euler scheme of multi-delay Caputoâ€“Fabrizio fractional-order differential equations,” *Appl. Math. Let.*, vol. 124, p. 107709, 2022.

[160] C. Lizama and M. Murillo-Arcila, “Discrete maximal regularity for volterra equations and nonlocal time-stepping schemes,” *Discrete Contin. Dyn. Syst. Ser. A*, vol. 40, no. 1, pp. 509–528, 2020.

[161] J. Prüss, *Evolutionary integral equations and applications*, vol. 87. Birkhäuser, 2013.

[162] C. Lizama, “The Poisson distribution, abstract fractional difference equations, and stability,” *Proc. Am. Math. Soc.*, vol. 145, no. 9, pp. 3809–3827, 2017.

[163] A. Kumar, J. Alzabut, S. Kumari, M. Rani, and R. Chugh, “Dynamical properties of a novel one dimensional chaotic map,” *Math. Biosci. Eng.*, vol. 19, no. 3, pp. 2489–2505, 2022.