

DEPICTING RANDOMNESS



AnDi CHALLENGE 2024

Braulio Fenollosa López
Violeta García Rodríguez
Aina Amelia Jiménez Benetó
Coral Montes López
Fanny Magdalena Muñoz Álvarez
Javier Palop Medina
Hugo Sánchez Navalón

CONTENT

INTRODUCTION.....	4
STATE OF ART	6
METHODOLOGY	7
SEGMENTATION AND CLASSIFICATION.....	10
REGRESSION	12
RESULTS	17
SEGMENTATION AND CLASSIFICATION.....	17
REGRESSION	17
DISCUSSION	21
LEGACY.....	22
ACKNOWLEDGMENTS	23
CONCLUSIONS	23
REFERENCES	25
APPENDIX	27
PHENOMENOLOGICAL MODELS.....	27
RANDOM FOREST.....	30

ABSTRACT

This study explores the dynamics of motion within single-particle experiments by analyzing various phenomenological models. Our primary goal is to uncover the fundamental mechanisms driving particle dynamics and understand their interactions with the environment. To achieve this, we focus on two primary tasks: segmentation and classification to identify behavioral shifts in particles, and regression analysis to predict diffusion coefficients and anomalous exponents for each trajectory segment.

A crucial aspect of our methodology is the computation of "absolute displacement" for each particle at each time step. This metric, which captures the sum of absolute changes in X and Y coordinates between consecutive time points, is vital for detecting changes in particle behavior. In the context of anomalous diffusion, "absolute displacement" helps mitigate noise and provides a clearer signal for identifying state transitions.

Utilizing machine learning models and rigorous validation techniques, we aim to refine our understanding of particle behavior and its environmental interactions. This research builds on findings from the first AnDi Challenge and leverages advanced methods such as statistical analysis, segmentation techniques, and machine learning algorithms. By developing new methods for analyzing individual particle trajectories, we contribute to the advancement of the field and enhance the understanding of anomalous diffusion phenomena in real-world scenarios.

INTRODUCTION

The world in our surrounding moves. No matter its scale, physical objects move in very particular ways, driven by their properties and their interaction with the environment. From the motion of black holes in the center of our own galaxy to the dynamics of particles in atomic experiments, their speed, direction and acceleration, among other properties, are widely used to understand their physical nature.

Inspired by the possibility of studying systems just by looking at their motion, we have elected to participate in the second edition of the Anomalous Diffusion Challenge where we will be evaluating methods to analyze motion changes in single particle experiments. In other words, we aim to analyze the behavior of single molecules to infer interaction kinetics and quantify their dynamic properties.

Initially, diffusion refers to numerous phenomena, by which particles and bodies of all kinds move throughout any kind of material. We envision a scenario where a particle, denoted as a "walker," undergoes random directional steps. When these steps exhibit displacement lengths following a Gaussian distribution with a specific variance, we classify this phenomenon as normal diffusion. In essence, the walker emulates the principles underpinning Brownian motion (*AnDi Challenge, 2024*).

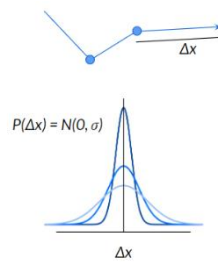


Figure 1. Brownian motion. The Mean Absolute Error of particles undergoing Brownian motion is directly proportional to both the diffusion coefficient (K) and time. The equation describing this motion is well-known, allowing the coefficient to be predicted from certain parameters or experimental conditions (Gegersen, 2024).

However, in real physical systems particles suffer deviations from this motion caused by the interaction with the environment. Particles may become trapped, bind with others, or transition between different modes, among other possibilities. We term any divergence from Brownian dynamics anomalous diffusion, meaning that the mean squared displacement is now not proportional to time but instead proportional to time at a certain exponent.

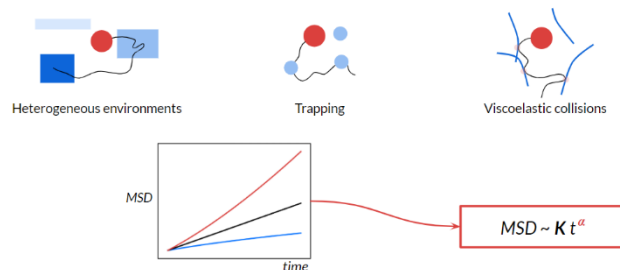


Figure 2. Caused by a deviation from normal diffusion. Now, the mean squared displacement is proportional to time at a certain anomalous exponent (α).

In our exploration of motion dynamics within single particle experiments, we delve into the realm of phenomenological models. These models capture the intricate interplay between a particle and its environment, shedding light on the diverse diffusion phenomena observed in real-world scenarios.

Central to our investigation are five distinct types of diffusion, each representing a unique manifestation of particle behavior (*AnDi Challenge, 2024*):

- **Single-state:** This model portrays trajectories characterized by consistent properties throughout their motion.
- **Multi-state:** Here, trajectories exhibit random transitions between different diffusive states.
- **Quenched trap:** In this model, particles undergo diffusion but can also become temporarily trapped within their environment.
- **Dimerization:** Considered within this model is the possibility of particles binding with one another if they come into proximity.
- **Confinement:** This model accounts for the confinement of particles within distinct compartments within their environment.

By examining these phenomenological models, visually depicted in *Figure 3*, we aim to decipher the underlying mechanisms driving particle dynamics and glean insights into the complex interplay between particles and their surroundings. Each model offers a unique perspective on diffusion phenomena, contributing to a comprehensive understanding of motion dynamics.

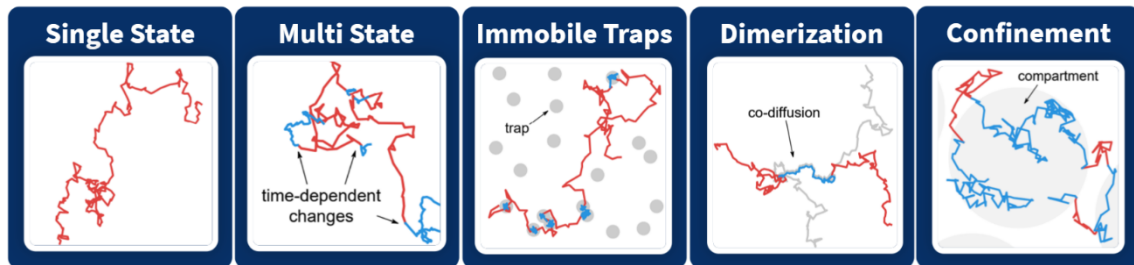


Figure 3. Phenomenological models.

To dive deeper into the context of the challenge, including details about datasets, metrics, and timelines, we strongly recommend reviewing the documentation provided by the organizers. Special attention should be given to the comprehensive manuscript titled "Quantitative Evaluation of Methods to Analyze Motion Changes in Single-Particle Experiments. (*Muñoz-Gil et al., 2023*)" Additionally, a seminar that serves as an end-to-end guide to the challenge has been broadcasted (*AnDi Challenge, 2024*). In this seminar, the organizers thoroughly explained the nature of the data (including videos and trajectories), the evaluation metrics, and provided guidelines for creating a proper submission. Access to both documents is available in the references section.

In this study, we delve into motion dynamics within single particle experiments by analyzing various phenomenological models. Our primary goal is to uncover the fundamental mechanisms driving particle dynamics and understand their interactions with the environment. To this end, we have formulated the following research questions: To what extent can the intrinsic properties of particle motion be characterized? Is it possible to accurately depict randomness?

We have identified two primary tasks that will guide the entire course of our investigation:

- Firstly, **segmentation and classification**, where we identify behavioral shifts in particles and extract corresponding segments along with their states.
- Secondly, **regression** analysis to predict diffusion coefficients and anomalous exponents for each trajectory segment.

Utilizing machine learning models and rigorous validation techniques, we aim to refine our understanding of particle behavior and its environmental interaction, contributing to broader scientific knowledge.

STATE OF ART

The field of particle diffusion has experienced a great deal of interest due to its fundamental role in various biological and physical processes. One specific area within this field is the analysis of anomalous motion and heterogeneity in the trajectories of individual particles. Proof of this boom is the AnDi Challenge 2, which seeks new methods to analyse particle motion, following the success of the first one. Our team is participating in this challenge, focusing on the analysis of single particle trajectories.

This research builds on both the results obtained in the first AnDi Challenge and previous work in this area, in particular the papers:

"Quantitative evaluation of methods to analyze motion changes in single-particle experiments" (Muñoz, 2023)

This article addresses the evaluation of methods to detect changes in the motion of single particles, with a focus on revealing heterogeneities and possibly anomalous diffusion in simulated environments inspired by biological systems. It builds on the results of the first AnDi Challenge and raises new issues, suggesting several methods as a starting point for our research. These methods include:

- Statistical methods: to identify patterns in particle trajectories.
- Segmentation techniques and algorithms: to divide a trajectory into segments representing different modes of motion, such as changes in velocity or direction.
- Machine learning algorithms: such as neural networks or support vector machines, to classify and analyse complex patterns in particle trajectories.

"Objective comparison of methods to decode anomalous diffusion" (Muñoz-Gil, 2021)

This article focuses on the objective comparison of methods to interpret anomalous diffusion. It analyses the results of the first AnDi Challenge and compares the performance of the methods presented by the participating teams in various tasks related to this phenomenon. Several methods stood out for their performance in decoding anomalous diffusion, showing consistent and compatible results, demonstrating their robustness and versatility. Most of these methods are based on machine learning architectures, such as recurrent neural networks (RNN), convolutional neural networks (CNN) or gradient boosting machines. Others are based on statistical approaches, such as Bayesian inference. Some methods employ feature engineering using classical statistics as input for machine learning.

In addition, we have tested and compared some of the methods described in:

"Characterization of anomalous diffusion through convolutional Transformers" (Wagner, T., 2017)

This paper presents the use of Convolutional Transformer (ConvTransformer), which combines convolutional neural networks to extract features from diffusive trajectories and transformers to perform regression or classification. It also employs convolutional and recurrent neural networks in

combination to achieve state-of-the-art results in anomalous diffusion exponent classification and regression.

[“Efficient recurrent neural network methods for anomalously diffusing single particle short and noisy trajectories” \(Orts, 2002\)](#)

This paper explores the use of recurrent neural networks (RNN), such as Long short-term memory (LSTM), to infer the anomalous exponent and classify the type of anomalous diffusion. In addition, it proposes the use of machine learning models, such as random forests and gradient boosting methods, to infer the anomalous exponent and classify trajectories into different diffusion models.

[“Classification and Segmentation of Nanoparticle Diffusion Trajectories in CellularMicro Environments” \(Firbas, N., 2023\)](#)

This paper describes the training of a Random Forest model with 9 different features to classify and segment individual trajectories into their respective motion types.

Our research leverages findings from the first AnDi Challenge and builds upon existing knowledge in this field. We draw inspiration from various methods, including statistical analysis, segmentation techniques, and machine learning algorithms like neural networks. This foundation is further enriched by exploring recent advancements in convolutional transformers and recurrent neural networks for classifying and analyzing anomalous diffusion.

By incorporating these various approaches, our research aims to develop new methods for analyzing individual particle trajectories. This will contribute to the advancement of the field and potentially have a significant impact on various scientific areas.

METHODOLOGY

The precise examination of statistical metrics concerning particle system dynamics hinges on obtaining precise, high-quality determinations of particle positions at sub-pixel levels. Yet, achieving optimal outcomes often demands tailored solutions for individual experiments. Although data-driven techniques have significantly advanced in surmounting the hurdles of designing tracking methods to specific datasets, the forefront of current methodologies heavily leans on the capacity to synthetically replicate the studied system. Given our goal of covering most of the phenomena typically encountered in real-world physical scenarios, creating a synthetic dataset is a complex undertaking.

To optimize our outcomes, we've chosen to harness the models offered by the challenge's organization. With this in mind, we'll construct our own dataset by delving into the singular class gathering all phenomenological models. It is important to mention that since the data are synthetically produced, the efforts in the data preparation phase have been oriented to the compression of the library and to gaining fluency in the manipulation of the functions rather than to the cleaning and treatment of missing data, since there are none.

Our approach involves examining these five distinct models wherein diffusion properties emerge both from the particle's interaction with its environment and from its inherent characteristics. The models we will work with are single state, multiple state, dimerization, immobile traps and confinement. We won't delve into the technical specifications of each model here, as those details can be found in [Table 1](#) in the [appendix](#). However, it is important to become acquainted with the common structure shared by all models and the specific events they aim to simulate.

All models have similar **inputs**, with some changing to accommodate the particularities of each. In general, we input the number of trajectories to be generated N , the number of time steps T , and then length of the box acting as environment L . To generate the trajectories, we can choose to either provide a fixed value for K and α (anomalous exponent), or a list containing a mean and a variance to sample the property from a normal distribution.

It's crucial to note that the diffusion parameters K and α , which define particle trajectories, exhibit characteristic ranges and unique patterns to each studied model. The diffusion coefficient K can take values from $[10^{-12}, 10^6]$ and the anomalous exponent α ranges from $[0, 2]$. The figure below shows an example of how distributed the diffusion parameters for each model are. Understanding these values can provide valuable insights into the type of phenomenon influencing the particle, aiding us in tackling our second task.

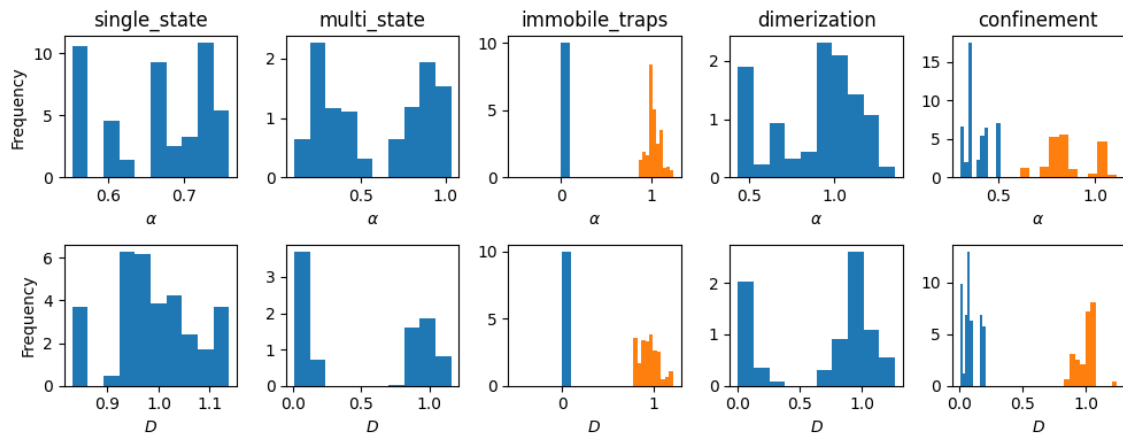


Figure 4. Diffusion parameters distribution for each model.

The **output** of all the phenomenological models is always the same: a tuple containing the trajectories and their labels. Each trajectory is represented as a list, with a list for each time instant containing pairs of coordinates. In other words, T lists are generated, each containing N sub lists representing the x and y position of each particle at a given time instant. The labels follow the same matrix structure, with T lists, each containing N internal lists. Within each internal list, we get the α , K , and a state label of a particle at a given instant. In Figure 4 we provide a schematic view of the input and output of the generated trajectories.

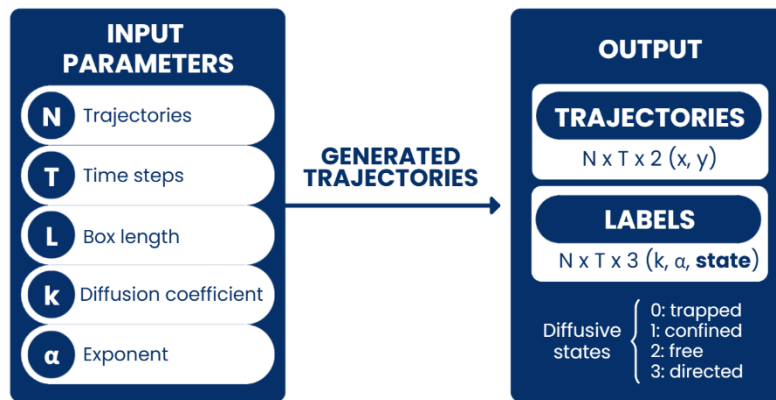


Figure 5. Schematic view of how the generation of the trajectories from the phenomenological models.

To ensure clear comprehension of our project, it's vital to distinguish between the **model** generating trajectories and the particle's **state** at any given moment. As previously mentioned, we employ five

models to generate trajectories, each representing different phenomena arising from particle-environment interactions. Conversely, there are four states in which a particle can be found: immobile (0), confined (1), free (2), and directed (3). The concept is that, along a trajectory generated by a particular model, the particle can transition between these states.

For instance, trajectories generated by the immobile model anticipate particles, typically in the free state (label = 2), transitioning to an immobile state (label = 0) at certain times. Conversely, those created from the confined model are expected to oscillate between the free and confined (label = 1) states. This distinction is crucial because detecting these shifts in labels between consecutive instances enables us to pinpoint change points and therefore, address our objective.

One final detail regarding our dataset structure that is noteworthy is that we have been provided with a class that defines default values for the challenge. By providing the diffusion model's number label, we can obtain a default dictionary containing the model's parameters. This dictionary can then be used to generate trajectories for the specifically selected model, which greatly facilitates our task.

To ensure clarity, we provide an illustrative code example:

```
dic = _get_dic_andi2(MODEL)

#MODEL takes one of the following values: 1,2,3,4,5 (each one
corresponding to a different model) and returns: default parameters

dic['T'] = T #Time instants

dic['N'] = N #Number of trajectories

trajs_p, labels_p = datasets_phenom().create_dataset(dics = dic)

trajs_p[i][j] = [x,y]

#Indicates the coordinates x and y for the particle j at the time i

labels_p[i][j] = [k, alpha, label]

#Indicates the diffusion coefficient, anomalous exponent and label
the for the particle j
```

This snippet illustrates the utilization of the provided class to access default parameters for a specific diffusion model and subsequently generate corresponding trajectories. It highlights the process of setting the number of time instants (T) and the number of trajectories (N), as well as how the trajectories and labels are stored, allowing for access to accomplish our tasks. This example provides a practical demonstration of our dataset creation process.

Now that we have a comprehensive understanding of the dataset, we can delve into the explanation of the tasks ahead.

The first task involves **segmentation and classification**. We begin by utilizing the previously generated trajectories as input and compute a new variable termed “absolute displacement”. To mitigate the noise introduced by anomalous diffusion, we create a softened version of this feature named “softened AD”, which involves constructing an ARIMA model. This allows us to treat our data as two temporal series, one for each coordinate. Subsequently, we can pinpoint instances where a particle undergoes a behavioral shift, transitioning from one state to another. This process of identifying changepoints facilitates the extraction of segments from each trajectory (segmentation)

and the assignment of corresponding states (classification). The efficacy of our approach will be assessed through the analysis of metrics such as the Jaccard coefficient and the RMSE, providing insights into the performance of our method.

For our next task, we maintain the particle trajectory coordinates as input, but we now focus on segments obtained individually. For these new inputs, we construct a dataset comprising 1500 trajectories, each containing 30 segments, generated by all five different models. This dataset ensures a balanced representation of trajectories from each model, thereby facilitating accurate results. Using this approach, we aim to predict the diffusion coefficient (K) and anomalous exponent (α) for each segment of the trajectory, effectively accomplishing the **regression** task.

To achieve this objective, we will assess several models, including MLP, SVM, RF and XGBoost. Initially, we will optimize each model by searching for hyperparameters that enhance their performance. Subsequently, for each model, we will conduct a k-fold validation to ensure robustness and reliability of the results. This rigorous process allows us to refine our models and validate their effectiveness in accurately predicting the diffusion coefficient and anomalous exponent for each segment of the trajectory.

Finally, to assess the precision of our predictions, we will compute the Mean Squared Log Error (MSLE) for the diffusion coefficient (K) and the Mean Absolute Error (MAE) for the anomalous exponent (α). These metrics will provide comprehensive insights into the accuracy and reliability of our predictive models.

The figure below illustrates a summary of the process undertaken to fulfill our objectives:

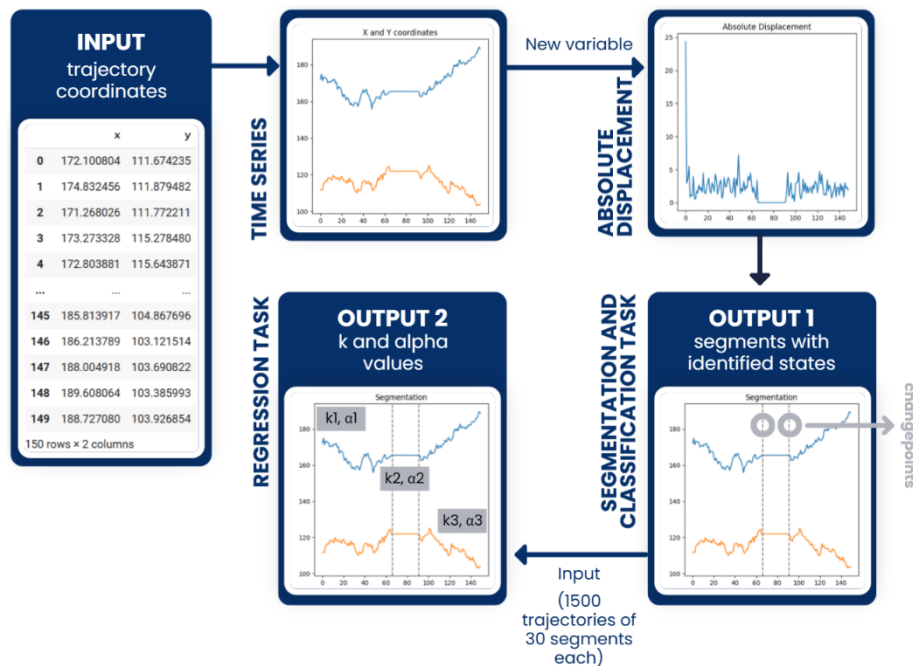


Figure 6. Minable view.

SEGMENTATION AND CLASSIFICATION

APPROACH

In our initial approach to trajectory segmentation, we aimed to predict the model that generated a specific trajectory and then apply corresponding segmentation rules. This leverages the fact that single-state, multi-state, and dimerization models primarily generate particles labeled “free”, while

confinement models produce trajectories with “confined” and “free” states, and immobile traps result in “immobile” or “free” labels. By identifying the generating model, we could automatically label all points in the first three categories as “free” and avoid unnecessary change point detection.

Unfortunately, accurately determining the generation model based solely on the resulting trajectory proved unreliable. Some models seem capable of generating very similar trajectories, hindering classifiers from learning the relationship between trajectories and their origin.

Given this limitation, we adopted a time-series-based approach. We created a new feature called “Absolute Displacement” (AD) for each particle at each time step. AD is calculated as the sum of the absolute changes in X and Y coordinates between consecutive time points. Notably, a change point signifying a transition from “free” to another state (“immobile” or “confined”, as “directed” is not observed in trajectories generated by any of the models provided by the organizers of the AnDi Challenge) will cause a significant variation in the AD trend. By applying some kind of set of thresholds in this trend, we might be able to determine the state changes.

However, directly analyzing AD can be misleading due to noise introduced by anomalous diffusion. To address this, we built a simple ARIMA model with a moving-average coefficient of 7 to smooth the AD values. This transformation creates a less volatile time series that allows for clearer analysis and conclusion drawing, which we will refer to as “softened AD”. The reason we chose 7 as the sliding window value is because of the threshold considered in the official metrics of the challenge with respect to the andi-datasets package criteria: when determining a change-point, 5 frames of margin are considered between ground truth and prediction as, because of anomalous diffusion, they might be non-deterministic (Muñoz-Gil 2024). Picking a 5 frames long window may result in a not enough softened trend description, while picking values much higher than 5 may result in ineffectiveness at determining actual changepoint. 7 seems a reasonable longitude that seems to maintain a desirable tradeoff between both considerations.

MODEL CONSTRUCTION

The states we want to predict are “immobile”, “confined” and “free”. By direct observation first and by studying generated softened AD time-series after, we can assert that “free” particles seem to consistently adopt values of softened AD higher than 1, while “immobile” ones just get stuck at 0. The values for “confined” ones tend to vary between 0 and 1. We have taken advantage of this to build a rule-based segmentation and classification model.

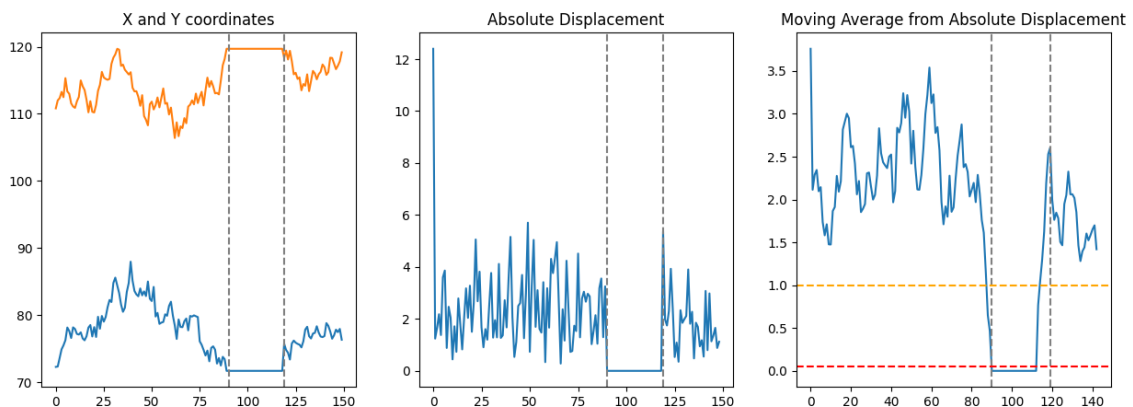


Figure 7. X and Y coordinates time series of a trajectory generated by immobile traps generator model. Corresponding Absolute Displacement time series time series. Corresponding softened AD with 7 frame-window time series with thresholds 0.1 (red) and 1 (yellow) marked. Ground truth change points are marked with discontinuous gray vertical lines.

As we can see in [Figure 7](#), and as we were previously saying, trajectories with alternating “immobile” and “free” states can be directly segmented by establishing a threshold on 0 in softened AD. As because of moving average nature there are 7 frames in which softened AD varies between 0 and 1, we will consider three first frames as of the old state and the remaining four as of the new one.

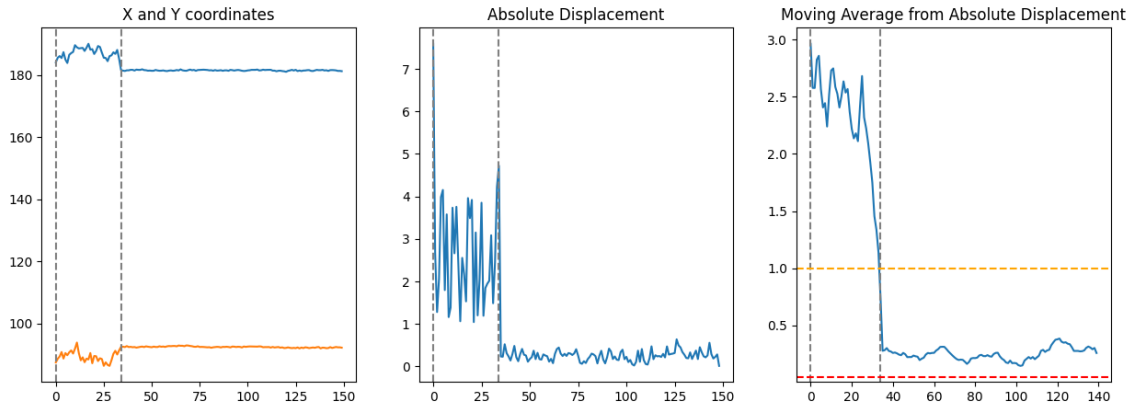


Figure 8. *X and Y coordinates time series of a trajectory generated by immobile traps generator model. Corresponding Absolute Displacement time series time series. Corresponding softened AD with 7 frame-window time series with thresholds 0.1 (red) and 1 (yellow) marked. Ground truth changepoints are marked with discontinuous gray vertical lines.*

We can see a similar behavior in particles with alternating “free” and “confined” states. In this case, as thresholds are based on a “soft” transition (there is no gap between intervals $[1, \infty[$ and $]0, 1[$), there is no need to rearrange “transition” softened AD values as we had to do in “immobile” - “free” transition scenario. We can just apply a threshold on 1 to determine whether a certain frame belongs to a “confined” or “free” segment and label as changepoint the frame in which a transition occurs.

All this means that our segmentation task for labeling each particle state can be reduced to calculating its softened AD-associated time series and classifying each frame as “immobile”, “confined” or “free” depending on whether it belongs to $\{0\}$, $]0, 1[$ or $[1, \infty[$ interval.

REGRESSION

The first step in our analysis is to create our dataset. Before delving into the details, it is essential to understand how particle states are generated. Each model can produce different states, but the “Free” state (2) can be generated by all models. Other states, such as “Immobile” (0), “Confined” (1), and “Directed” (3), are specific to a limited number of models (one or two).

When creating the synthetic dataset, our primary goal was to achieve the most balanced data distribution possible for efficient model training. We explored two key approaches to achieve this:

1. **Model-based Balancing:** In this approach, data is generated with balanced representation from each generation model.
2. **State-based Balancing:** This approach prioritizes balanced representation based on specific particle states within the generated data.

While both approaches share the foundation of generating data with the same parameters, their focus on balancing differs. The first option involves creating a general training dataset without explicitly ensuring balanced label distribution across all states. This approach includes data from all generation models (1 to 5) in (potentially) equal proportions within the final output (X and Y). The second one represents an improved approach that focuses on achieving a balanced number of samples for each targeted state (identified by the label information, Y).

Choosing model-based balancing has advantages, such as general representation of data from all models, robustness to variations between models, and greater generalization. However, it also presents disadvantages, including bias towards more frequent states (such as the "Free" state in this case) and lower performance in specific states.

On the other hand, state-based balancing offers balanced representation of specific states (prioritization of critical states), lower bias, and greater accuracy in relevant states. However, it may have a potential impact on generalization to unseen data, meaning the model may be less versatile and applicable in real-world scenarios where data may come from various sources.

The choice between the two depends on the objectives. Model-based balancing may not be ideal when certain types of states are rare or crucial for specific analysis. Therefore, we opted to use data balanced according to the state, as it provided less bias. This result was foreseeable, as balancing according to the generating model resulted in an imbalance with a greater number of samples in the Free state (2), because all models generate this state.

The state-based balancing approach ensures that the dataset used to train the model is representative and balanced, thus improving the efficiency and accuracy of the trained model.

Once the balanced dataset with the desired number of trajectories and their lengths has been created, this task, whose main goal is to predict the diffusion coefficient K and the anomalous exponent α , will be carried out using various regression models, including SVM, RF, XGBoost and MLP. The reason for using multiple regression models is to enable subsequent comparison. The goal is to evaluate whether all the models behave the same way across all ranges of the predicted variable values. Additionally, it aims to analyze if the complexity of each model affects the prediction accuracy. By comparing different models, it is possible to identify which one offers better performance and to understand the differences in their behavior and evaluation. The selection of hyperparameters for each model was performed through an exhaustive search using the scikit-learn library. For each dataset and each parameter, a range of values was explored, and those that maximized predictive performance on the test set were selected. The optimal values selected are presented along with the results of each experiment.

Finally, the ability of each model to generalize to unseen datasets and different experimental conditions needs to be evaluated. Therefore, in the case of the dataset with the input variable transformed by absolute displacement, we conducted a k-fold cross-validation with 5 partitions to assess the robustness of our model. The k-fold cross-validation involves dividing the dataset into k subsets (folds) of approximately equal size. Then, the model is trained k times, each time using $k-1$ folds as training data and the remaining fold as test data. This strategy allows for efficient use of all available data for both training and evaluation and reduces the risk of overfitting to the model. This technique is essential to ensure the generalization and stability of model performance across different partitions of training and test data.

The subsequent section delves deeper into the specifics of the models used, including implementation details, performance evaluations, and comparisons of:

1. Support Vector Machines
2. Multilayer Perceptron
3. XGBoost
4. Random Forest

This analysis offers a nuanced understanding of the strengths and limitations of each model, critically examining their performance under diverse experimental conditions.

SVM

Support Vector Machines (SVMs), developed by Vapnik and his team, are powerful tools for supervised learning, particularly effective in tasks like optical character recognition (OCR) ([Smola & Schölkopf, 2004](#)). SVMs are divided into Support Vector Classification (SVC) and Support Vector Regression (SVR). SVR, introduced by Vapnik, Golowich, and Smola in 1997, focuses on regression tasks by minimizing generalized error bounds, which combine training error and a regularization term ([Basak et al., 2008](#)). This approach ensures robust performance, especially in noisy and non-linear data scenarios. SVMs use kernel functions (e.g., linear, polynomial, radial basis function) to transform input space, making non-linear data linearly separable, which is crucial for capturing complex relationships.

A notable application of SVM in trajectory analysis was conducted by [Helmuth et al. \(2007\)](#). They developed an SVM-based trajectory segmentation algorithm for identifying trajectory fingerprints of adenovirus particles in live cells. This study demonstrated SVM's capability to handle synthetic data and segment different types of motion, supporting its potential in our regression tasks.

To identify the optimal hyperparameters for SVR, we employed Halving Grid Search. This systematic search method narrows the hyperparameter space by evaluating fewer candidates in each iteration while dedicating more resources to the most promising ones. This approach balances thorough exploration with computational efficiency, ensuring a good balance between model complexity and performance, and avoiding overfitting while achieving high predictive accuracy. For our SVR model, the optimal hyperparameters identified were {'C': 1, 'epsilon': 0.01, 'gamma': 0.01, 'kernel': 'rbf'}, providing a balanced and robust performance ([Ghosh, 2024](#)).

MLP

Multi-layer perceptrons (MLPs) are fundamental neural network architectures known for their remarkable versatility. They can handle a wide range of machine learning tasks, including classification, regression, image recognition, and natural language processing. In regression tasks, such as ours, neural networks are particularly valuable for their ability to model complex relationships between inputs and continuous outputs and specially excel at capturing non-linear dependencies and interactions within the data, leading to highly accurate predictions ([aamirkhthgf Follow, 2023](#)).

The Anomalous Diffusion Challenge (AnDi Challenge) demonstrated that machine learning methods, particularly neural networks, can outperform classical statistical methods in characterizing anomalous diffusion. In AnDi Challenge 1, a bi-layered transformer based convolutional neural network, ConvTransformer, achieved the best performance in many cases, highlighting its effectiveness in both inferring the anomalous diffusion exponent (Task 1) and determining the underlying diffusive regime (Task 2). Inspired by these results, we are now adapting the MLP model to our regression task in AnDi Challenge 2. As another neural network-based model sharing the fundamental principle of learning from data using multiple layers of neurons, we aim to harness its proven capabilities to effectively meet our objectives. ([Firbas et al., 2023](#))

For our task, we used the MLPRegressor from the `sklearn.neural_network` library. The training involves forward propagation, loss calculation, and backpropagation, repeated over several iterations. Extensive grid search identified optimal hyperparameters: an activation function of 'tanh', a hidden layer size of 200 neurons, and the 'adam' solver were found to be the best combination. These parameters were chosen for their balance between model complexity and performance. The 'tanh' activation function, or hyperbolic tangent, helps in capturing non-linear relationships. A hidden layer size of 200 neurons strikes a balance between underfitting and overfitting, and the

'adam' solver, a stochastic gradient-based optimizer proposed by Kingma and Ba, ensures efficient and effective training. (*Pedregosa et al., 2011*)

XGBOOST

Scalability is a crucial consideration when designing systems to accommodate growing demands. As the project progresses and we officially participate in the challenge, the number of trajectories we need to train will increase significantly. To address this need for scalability, we have implemented XGBoost.

XGBoost, or eXtreme Gradient Boosting, stands out for its scalability and performance in supervised machine learning. It optimizes by minimizing a differentiable loss function through gradient descent, enhancing predictions iteratively with decision trees. Each tree's weighted score contributes to the final prediction, while a regularized objective function prevents overfitting. (*Chen, T., & Guestrin, C.*)

Numerous studies underscore XGBoost's adaptability and efficacy across diverse applications. For example, *Garibo-i-Orts et al.* employed XGBoost to classify normal, super-, and subdiffusion trajectories, demonstrating its versatility. Similarly, *Firbas et al.* utilized XGBoost to characterize anomalous diffusion via convolutional transformers. In comparisons of methods for decoding anomalous diffusion, XGBoost and Gradient Boosting were employed for feature extraction and classification tasks, highlighting XGBoost's robustness and flexibility.

For our regression task, XGBoost was a natural choice due to its reputation as a potent and efficient learning algorithm. Leveraging default hyperparameters, meticulously chosen for a balance between model complexity and performance, ensures our model handles complex datasets effectively. Parameters like {'Booster': 'gbtree', 'Eta': 0.3, 'Gamma': 0, 'Max Depth': 6, 'Min Child Weight': 1, 'Lambda': 1, 'Alpha': 0} provide a robust foundation for efficient and accurate modeling. These default settings are strategically designed to reduce overfitting, with 'Eta' controlling the learning rate, 'Gamma' enforcing a minimum loss reduction for splitting nodes, 'Max Depth' limiting tree depth, 'Min Child Weight' ensuring a minimum number of samples in each leaf node, and 'Lambda' and 'Alpha' adding regularization to the weights to prevent the model from becoming overly complex. (*Dmlc. (s. f.)*)

RF

Considering the synthetic nature of our data and the opaque characteristics of other models explored, we have opted to employ a Random Forest (RF) regression model for predicting α and k . RF, an ensemble learning method, is particularly well-suited for this task due to its intrinsic ability to manage overfitting through the integration of multiple decision trees. This integration significantly bolsters its ability to generalize across diverse datasets, ensuring robust model performance.

The model operates by constructing multiple decision trees during the training phase, with each tree built from a randomly sampled subset of the data, promoting diversity and reducing bias (*Breiman, 2001*). This process, known as bagging, involves selecting a subset of elements from the dataset (with replacement) and only a fraction of the total number of features to split the nodes, enhancing the model's robustness to overfitting. The ensemble of trees then aggregates their individual predictions to form a final forecast, typically an average. This approach harnesses collective insights over singular interpretations, thereby enhancing prediction accuracy and stability across varied datasets (*Wagner et al., 2017*).

In our literature review, Random Forest has proven highly effective for both classification and regression tasks in particle diffusion studies. For instance, *T. Wagner et al.* highlighted RF's ability to

classify and segment nanoparticle diffusion trajectories with high precision. Similarly, research by [Nicolás Firbas et al. \(2023\)](#) demonstrated RF's versatility in predicting the anomalous diffusion exponent and classifying diffusion types. These studies underscore RF's robustness in handling complex data tasks, prompting us to consider whether adapting RF to our specific trajectory data could yield similar success in regression challenges.

The model was implemented using the `sklearn` in Python. Subsequently, we engaged in hyperparameter tuning, a process aimed at optimizing the model's settings to enhance performance ([Koehrsen, 2017](#)). After efficiently exploring the parameter space, the optimal combination both for α and k models were: `{'criterion': 'squared_error', 'max_depth': None, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 500}`

The selected configuration for the model strategically emphasizes precision and robustness. It employs 'squared_error' as the criterion to minimize prediction errors and allows unrestricted tree depth to capture detailed data features comprehensively. Furthermore, using 500 estimators strikes a balance between computational efficiency and the capability to robustly generalize complex patterns in the dataset, thereby enhancing model accuracy and reliability.

Now that we have detailed the implementation, it is crucial to explore what sets this model apart from others: its **interpretability**. A key question emerges: How does the model derive its values? To unmask the workings of the RF, we can employ two approaches: examining a single tree within the forest and analyzing the feature importances of the explanatory variables.

Focusing on the information contained by one tree we can observe the decision-making process, identifying which variables are considered and where segmentations occur. To facilitate understanding, we will limit the depth of the trees, allowing us to produce a more comprehensible visual representation of the tree's structure.

We have taken the first tree of α 's RF model as an example, which can be consulted in [Figure 12](#) of the [Appendix](#). As we traverse each branch, we observe that decisions at various nodes split the trajectories based on the values at specific time instants, leading to distinct predictions of α . While the individual variables might not immediately convey a clear semantic meaning, the visualization notably demonstrates how new trajectory parameters can be estimated. This process is effectively reduced to answering a series of simple, binary questions at each node, which guides us through the decision paths to arrive at a prediction, facilitating an intuitive understanding of predictive dynamics.

The second approach to grasp interpretability involves quantifying the usefulness of all the variables in the entire random forest by looking at the relative importances of the variables. In our specific context, the variables represent the absolute displacement in discrete time instants, which complicates direct interpretation.

However, upon analyzing the feature importances within our model, it is noteworthy that the earlier segments of the trajectories—specifically, the first half of the displacement data—consistently exhibit a higher level of importance, what can be found in [Table 2](#) of the [Appendix](#) section. This observation prompts us to question whether these early instants inherently carry more significant information about the motion characteristics or if this is a modeling artifact. To delve deeper, we generated trajectories of varying lengths and confirmed that this pattern persists. This finding suggests that the model heavily relies on initial behaviors to inform its predictions. We will consider this insight during the final phase of the challenge, ensuring enhanced model performance through strategic feature selection.

RESULTS

SEGMENTATION AND CLASSIFICATION

As the method used has been asserted throughout direct observation and case-studying rather than by machine learning inference, there is not a metric-minimization process involved nor a real possibility to build a cross-validation schema. However, it is possible to measure its performance. To do that, we will consider Jaccard index and RMSE.

We use Jaccard index to check for correct changepoint detection. It is calculated as the ratio of True Positives (changepoints marked in the range determined by the threshold) to the sum of True Positives, False Negatives (changepoints not detected) and False Positives (changepoints marked which are not labeled as so by validation data).

RMSE, or Root Mean Squared Error, is then used over those changepoints labeled as True Positives. It is defined as the root of the mean of the square of the frame's deviation of the detected changepoint with respect to the validation data provided.

We ran a 1500-trajectories experiment in which we applied the model previously built. Trajectories were extracted from a dataset generated by equal representation of all five generator models. We got a mean Jaccard index of 1 with 0 standard deviation, which means that all changepoints were correctly detected.

Then, after measuring RMSE over all our detected changepoints for each trajectory, we got the distribution shown in [Figure 9](#).

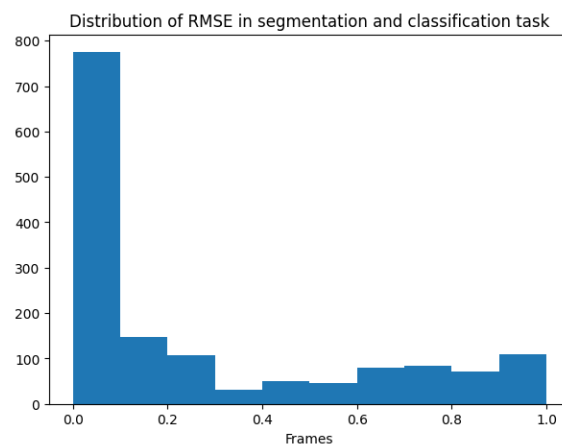


Figure 9. Visualization showing RMSE distribution in all 1500 trajectories segmented by the rule-based model built.

As we can see, changepoints from more than half of the trajectories considered were detected within less than a 0.1 frame deviation. None of the trajectories considered gave out a RMSE in detection of more than one frame, with global RMSE being 0.4275.

REGRESSION

In our evaluation of the regression task, we focus on two key metrics: the Mean Squared Log Error (MSLE) for the diffusion coefficient (K) and the Mean Absolute Error (MAE) for the exponent alpha (α). These metrics serve as crucial indicators of the accuracy and reliability of our regression models in predicting the dynamic properties of particle motion.

Figure 10 provides a clear visualization of the MAE and MSLE values for both parameters across all models. These figures offer an accessible comparison, allowing for a straightforward assessment of model performance in capturing the intricacies of particle behavior

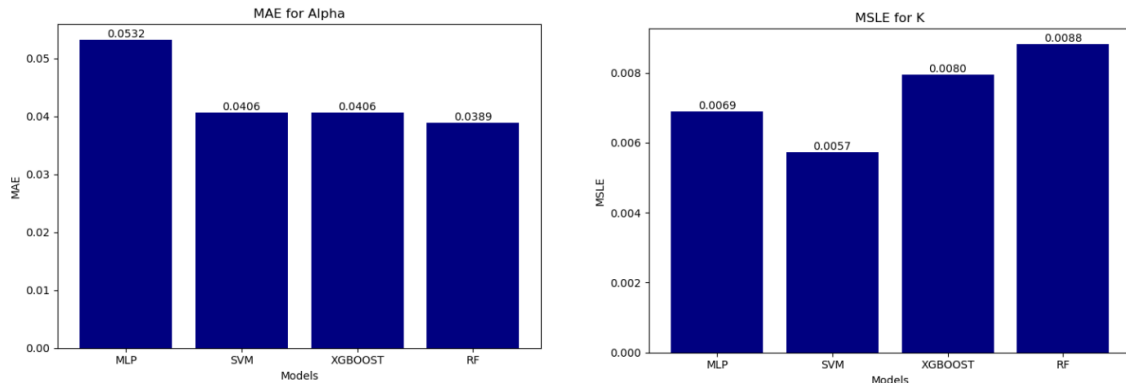


Figure 10. MAE value for alpha and MSLE for k across every model.

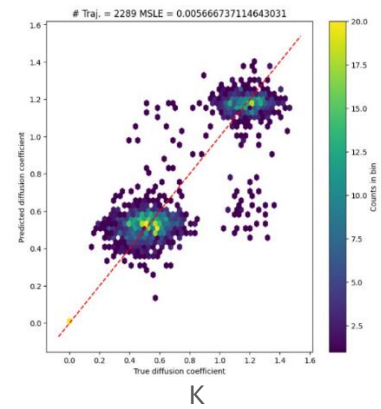
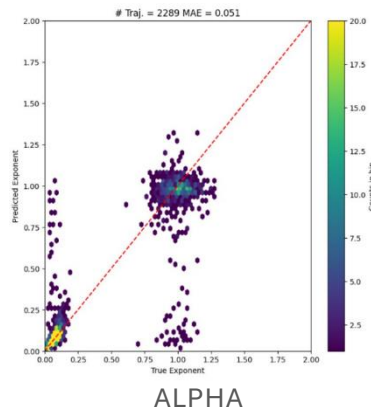
Upon analysis, it becomes evident that the Mean Absolute Error (MAE) values for the exponent alpha are consistently low across all models. Notably, the MAE ranges from 0.0532 for MLP to 0.0389 for RF, showcasing a remarkable precision in predicting alpha. This uniformity in low MAE values underscores the efficacy of our regression models in accurately estimating this critical parameter, essential for comprehending particle dynamics.

Conversely, upon scrutinizing the Mean Squared Log Error (MSLE) values, we detect minor discrepancies among the models. While all models demonstrate relatively low MSLE values, RF, which boasted the lowest MAE value, exhibited a slightly higher MSLE than the others. Despite this slight variance, it's crucial to note that the disparities in MSLE values between the models remained marginal, indicating a consistent and robust performance across the entire range of models.

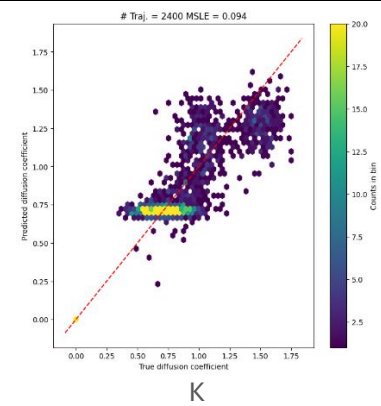
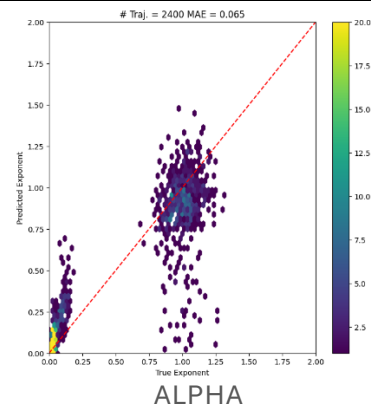
These results highlight the non-uniform performance of models across different predictive tasks. Notably, RF excels in predicting Alpha but performs relatively poorer in estimating K. This variability underscores the need for a nuanced approach to model selection, tailored to the specific prediction task at hand. Moreover, the synthetic nature of the data used for these tests necessitates careful consideration. Models that perform well with synthetic data may behave differently when applied to real-world data in a competitive setting. Therefore, the reliability and robustness of each model must be thoroughly evaluated with real data before making a definitive choice for the challenge. This ensures not only optimal performance but also adaptability and accuracy in practical applications.

In summary, our regression analysis yielded promising results, with all models demonstrating commendable accuracy in predicting both the diffusion coefficient and the anomalous exponent alpha. Nevertheless, to gain further insights into our results, we analyze the distribution of the predicted values alongside the real ones. This approach allows us to pinpoint areas where our models may be prone to flaws or exhibit higher prediction errors, thus providing valuable context for understanding the nuances of our regression performance.

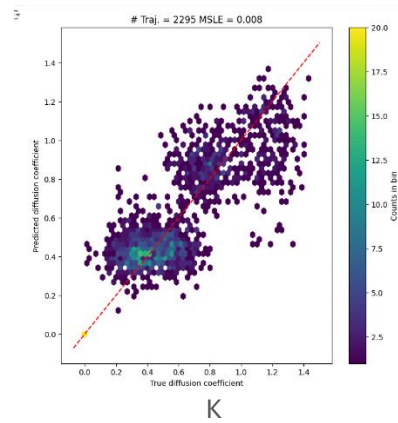
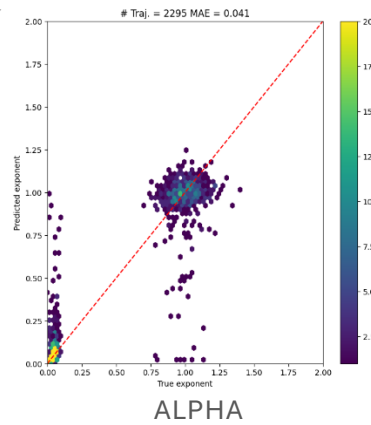
SVM



MLP



XGBOOST



RF

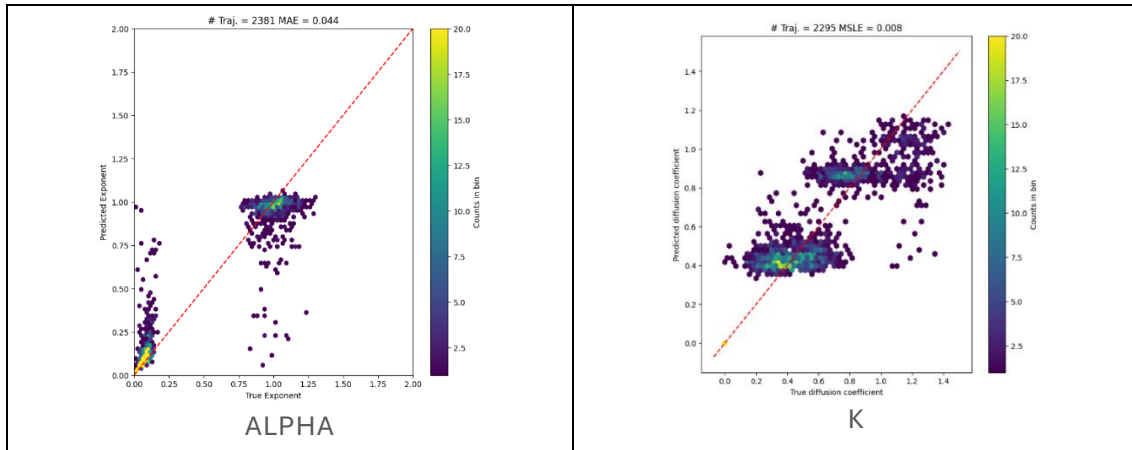


Figure 11. Error distribution for the parameters *alpha* and *k* for every model

ALPHA PREDICTION

Generally, the plots indicate an alignment between the predicted and true values of Alpha, with a fair concentration of data points around the diagonal line across all models. However, there is notable dispersion, especially for lower values of the true exponent, suggesting some challenges in accurately predicting smaller Alpha values.

Among the models, XGBoost stands out with a dense clustering near the diagonal, reflecting its high accuracy and robust performance in predicting Alpha. Random Forest also shows a good correlation, albeit with some variability. In contrast, MLP displays the weakest alignment, with a broader dispersion of data points, indicating lower accuracy compared to others. Observations that align to the performance shown in terms of MAE in the previous graphs.

In addition to assessing model performance, these plots prompt reflection on the synthetic data generation process. The high concentration of data points at zero suggests a significant number of trajectories where the anomalous exponent implies no motion, which could be an artifact of how the synthetic data are structured or an indication of model limitations in capturing more dynamic behaviors.

K PREDICTION

The predictions for K generally show a denser and more cohesive clustering of points along the diagonal compared to Alpha predictions, indicating a closer match between predicted and true values across models. This suggests that models are generally more accurate in predicting K.

MLP and SVM exhibit strong performance, with MLP maintaining precision across a range of K values and SVM showing strong model performance despite some scatter. Random Forest, while effective, displays occasional inaccuracies with a broader dispersion of points.

The plots suggest that models tend to perform better in predicting K than Alpha, with the real values of K being better distributed across the plots. For both SVM and Random Forest, there is an apparent bias in the distribution of real K values, evidenced by two distinct clusters of points. This may reflect nuances in the synthetic data generation or inherent model biases that could influence performance.

Overall, these insights highlight the strengths and areas for improvement in each model and emphasize the importance of model selection and tuning based on specific predictive needs and data characteristics.

Furthermore, we examined the Coefficient of Determination (R^2) values for both Alpha and K among the four models under study. These analyses revealed high predictive accuracy, suggesting that the models adeptly capture the variance present in the dataset.

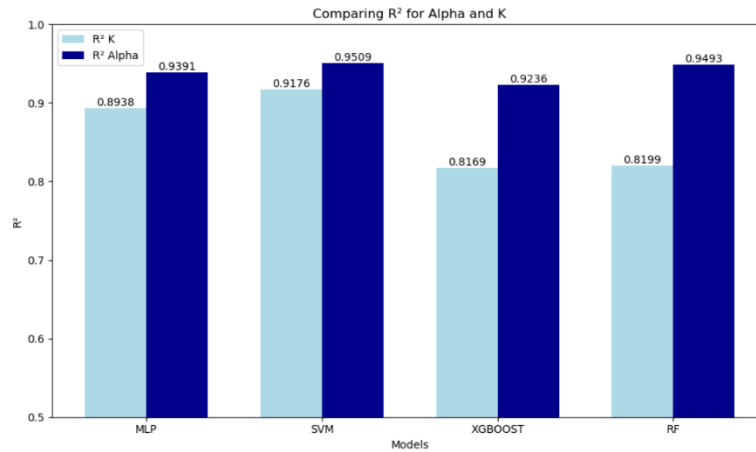


Figure 12. Comparison of the R -squared values for the 4 studied models.

Notably, the R^2 values for Alpha (ranging from 0.92 to 0.95) are consistently higher than those for K (ranging from 0.81 to 0.91), suggesting a better fit for predicting the anomalous exponent Alpha. This could be due to the fact that Alpha tends to have more distinct and stable patterns that models can capture more effectively, as it only ranges between values of 0 and 2. On the other hand, K, the diffusion coefficient, can be influenced by a wider range of factors, including environmental interactions and particle-specific properties, leading to greater variability and complexity. This inherent variability makes K more challenging to predict accurately.

Among the models, SVM achieves the highest R^2 for both K (0.9176) and α (0.9509), indicating that SVM is particularly effective at capturing the underlying patterns in the data for both diffusion parameters. RF also performs well, particularly for the diffusion coefficient K with an R^2 of 0.9493. However, its performance drops significantly for predicting Alpha, with an R^2 of only 0.8199. This stark contrast suggests that RF struggles to capture the nuances of Alpha compared to its performance in predicting K.

These findings underscore the nuanced dynamics of prediction tasks and highlight the model's varying strengths and limitations in capturing diffusion parameters effectively.

DISCUSSION

With respect to **segmentation and classification tasks**, we can assert that our rule-based model, by correctly determining all changepoints for every trajectory generated, catches up with state-of-the-art classification methods. In the case of solely considering segmentation problem, our rule-based model seems well calibrated as, for every trajectory generated, its mean deviance stays below 1 frame.

Given those results, we see that little can be done to further improve our results considering only synthetic data. However, ground-truth data available at validation phase at AnDi Challenge might provide some useful insights to consider in order to fine-tune it to challenge requirements.

In the **regression task**, we explored two main strategies for generating balanced datasets and compared their results in terms of the predictive capability of the models. Firstly, a dataset balanced by the states of the particles at each instant was created. This allowed for a balanced representation of the different behaviors of particles during anomalous diffusion (immobile, confined, free, and directed). On the other hand, a dataset balanced by the five trajectory generation models used was generated. This strategy allowed for an equitable distribution of trajectories generated by each model, providing a more comprehensive representation of the different types of diffusion behaviors (single-state, multi-state, immobile-traps, dimerization, and confinement).

We observed that the predictive performance of our models varies significantly depending on the type of dataset used. In the first approach, where data was balanced by particle states, results were suboptimal, with R^2 values close to zero or even negative. This suggests that information about particle states may not be sufficient to capture the complexity of anomalous diffusion.

On the other hand, balancing the dataset by trajectory generation models showed a slight improvement in predictive performance. However, the results were still unsatisfactory, indicating that variability between generation models may not be the most relevant feature for predicting diffusion parameters. Additionally, it is important to note that while the approach of balancing by trajectory generation models produced better results than the previous dataset, there is a risk of introducing biases into the model due to the unequal distribution of states of trajectories generated by each model. That's the reason why we think it is a better option the dataset balanced by states than this one.

We then investigated an alternative approach by transforming the input variable, which initially consisted of the raw coordinates (x , y) obtained from each segment of the trajectories. This transformation involved calculating the absolute displacement between successive data points, leading to a notable improvement in predictive accuracy. Both the MAE and MSLE scores indicate significantly enhanced performance in predicting the K and α parameters, respectively. This method appears to better capture the dynamics of anomalous diffusion by directly considering the magnitude of displacements between successive data points.

Moreover, k -fold cross-validation results for the input variable transformed by absolute displacement demonstrate consistent and robust performance across different data partitions, suggesting strong generalization capability to new datasets. Hence, the transformation of the input variable has substantially enhanced the model's predictive capacity, offering promising outcomes for estimating α and K parameters within the context of anomalous diffusion.

Overall, our results demonstrate promising performance, largely attributed to our utilization of synthetic data. However, it's important to acknowledge that these findings may be subject to alteration when working with larger, more diverse datasets. While synthetic data provides a controlled environment for method evaluation, the true test lies in applying our methods to real-world data, such as that provided by the Anomalous Diffusion Challenge. The dataset provided by the challenge is significantly larger and more representative of real-world scenarios, offering a more rigorous assessment of our methods. Therefore, to obtain a more comprehensive understanding of our method's effectiveness, we plan to train our models using the challenge's data and evaluate the results on the platform to obtain real error values. This will provide valuable insights into the performance of our methods in practical applications.

LEGACY

Upon completion of the project, we have devised a comprehensive plan to disseminate our findings, code, and data to ensure accessibility and encourage further research in the scientific community.

Our codebase, along with the datasets used and the project report, will be made publicly available on a designated repository, such as GitHub, following the conclusion of the Challenge. This will

facilitate replication of our study and provide a valuable resource for researchers interested in exploring similar phenomena. Additionally, thorough documentation will be provided to guide users through the code implementation and data analysis process, enhancing replicability and ease of use.

Furthermore, we envision our project as a springboard for future research endeavors. By sharing our insights and methodologies, we aim to inspire and empower other scientists to build upon our research and explore new avenues in the field of particle dynamics.

In terms of societal impact, our project aligns with the Sustainable Development Goals (SDGs) outlined in the Agenda 2030. This study contributes to Target 9.5 by advancing scientific research and promoting innovation in particle dynamics analysis. Additionally, our research supports Target 3.B.2 by fostering the development of medical research and basic healthcare through the generation of new knowledge and insights.

Moreover, our work has implications for various other SDGs. By providing valuable insights into particle movement and dynamics, our research can inform climate change mitigation strategies (SDG 13), contribute to biodiversity conservation efforts (SDG 14 and SDG 15), and promote industry, innovation, and infrastructure development (SDG 9).

Overall, our project serves as a foundation for further scientific inquiry and has the potential to drive meaningful advancements in particle dynamics analysis, ultimately contributing to the broader goals of sustainable development and societal progress.

ACKNOWLEDGMENTS

We extend our sincere gratitude to Professor Alberto from the Universitat Politècnica de València (UPV) for his invaluable guidance and support throughout the duration of this project. His expertise, encouragement, and dedication have been instrumental in helping us navigate the complexities of our research and overcome challenges along the way. We are deeply grateful for his mentorship and for providing us with the opportunity to pursue this project under his supervision.

CONCLUSIONS

Our comprehensive analysis delves into two critical aspects of particle dynamics analysis: segmentation/classification and regression. Through rigorous evaluation, we have demonstrated the efficacy of our models in capturing the intricate behaviors of particles and predicting key parameters governing their motion.

In segmentation and classification, our rule-based model exhibited exceptional performance, accurately detecting all changepoints with a mean Jaccard index of 1. This underscores its competitiveness with state-of-the-art classification methods. Moreover, the negligible RMSE values indicate precise changepoint detection across trajectories, highlighting the model's robustness.

Regression analysis further reinforced our findings, revealing commendable accuracy in predicting the diffusion coefficient (K) and the anomalous exponent α across various models. While minor discrepancies were observed, particularly in the predictive capability of different models, overall performance remained consistently strong. Notably, the transformation of input variables led to substantial improvements, enhancing predictive accuracy and generalization capability.

Our investigation also shed light on the importance of dataset balancing and input variable transformation in optimizing model performance. While balancing by particle states initially yielded

suboptimal results, the transformation of input variables significantly enhanced predictive accuracy, emphasizing the need for nuanced approaches to data preprocessing.

Looking ahead, our findings lay a solid foundation for future research endeavors. By disseminating our insights, methodologies, and codebase, we aim to facilitate replication and inspire further exploration in the scientific community. Moreover, our project's alignment with the Sustainable Development Goals underscores its potential to drive meaningful advancements in particle dynamics analysis, with implications spanning various fields, from healthcare to environmental conservation.

In essence, our study represents a significant step forward in understanding particle dynamics, offering valuable insights and methodologies that pave the way for continued progress and innovation in this crucial area of research.

REFERENCES

- Muñoz-Gil, G., Bachimanchi, H., Pineda, J., Midtvedt, B., Lewenstein, M., Metzler, R., Krapf, D., Volpe, G., & Manzo, C. (2023). Quantitative evaluation of methods to analyze motion changes in single-particle experiments. En arXiv [cond-mat.soft]. <http://arxiv.org/abs/2311.18100>
- Breiman, L. (2001). Machine learning, 45(1), 5–32. <https://doi.org/10.1023/a:1010933404324>
- Wagner, T., Kroll, A., Haramagatti, C. R., Lipinski, H.-G., & Wiemann, M. (2017). Classification and segmentation of nanoparticle diffusion trajectories in cellular micro environments. PloS One, 12(1), e0170165. <https://doi.org/10.1371/journal.pone.0170165>
- Koehrsen, W. (2017, diciembre 27). Random forest in python. Towards Data Science. <https://towardsdatascience.com/random-forest-in-python-24d0893d51c0>
- Firbas, N., Garibo-i-Orts, Ò., Garcia-March, M. Á., & Conejero, J. A. (2023). Characterization of anomalous diffusion through convolutional transformers. Journal of physics. A, Mathematical and theoretical, 56(1), 014001. <https://doi.org/10.1088/1751-8121/acafb3>
- Orts, Ò. G. i., Garcia-March, M. A., & Conejero, J. A. (2021). Efficient recurrent neural network methods for anomalously diffusing single particle short and noisy trajectories. En arXiv [cs.LG]. <http://arxiv.org/abs/2108.02834>
- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. En arXiv [cs.LG]. <http://arxiv.org/abs/1603.02754>
- Meyer-Base, A., & Schmid, V. J. (2014). Pattern recognition and signal analysis in medical imaging. Academic Press.
- aamirkhthgf Follow, A. (2023, octubre 12). Multi-layer perceptron a supervised neural network model using sklearn. GeeksforGeeks. https://www.geeksforgeeks.org/multi-layer-perceptron-a-supervised-neural-network-model-using-sklearn/?ref=ml_lbp
- Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
- Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).
- Helmuth, J. A., Burckhardt, C. J., Koumoutsakos, P., Greber, U. F., & Sbalzarini, I. F. (2007). A novel supervised trajectory segmentation algorithm identifies distinct types of human adenovirus motion in host cells. Journal of Structural Biology, 159(3), 347–358. <https://doi.org/10.1016/j.jsb.2007.04.003>
- Basak, D., Pal, S., & Patranabis, D. C. (2008). Support Vector Regression. https://www.researchgate.net/publication/228537532_Support_Vector_Regression
- Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. Statistics and Computing, 14(3), 199–222. <https://doi.org/10.1023/b:stco.0000035301.49549.88>
- AnDi Challenge 2024. (2024, February 22). AnDi Challenge. <http://andi-challenge.org/challenge-2024/>. Seminar available at:

https://docs.google.com/presentation/d/1uAd-hYHdLZpx3v-PJHL0Z6DhaJzse06gz_2wOwnb1qY/edit#slide=id.p

Gregersen, Erik. (2024, May 2). The Editors of Encyclopedia Britannica. (2024). Brownian motion. En Encyclopedia Britannica. <https://www.britannica.com/science/Brownian-motion>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2012). Scikit-learn: Machine Learning in Python. In arXiv [cs.LG] (pp. 2825–2830). <https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>

Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241-259.

Breiman, L. (1996). Stacked regressions. Machine Learning, 24(1), 49-64.

Ghosh, S. (2024, April 13). Hyper parameter tuning techniques — grid search, Bayesian & halving— wonders of ML realm. Medium. <https://medium.com/@ghoshsiddharth25/hyper-parameter-tuning-techniques-grid-search-bayesian-halving-wonders-of-ml-realm-45d6e2e73440>

Sklearn.Svm.SVC. (n.d.). Scikit-Learn. Retrieved May 21, 2024, from <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

Muñoz-Gil, G., Volpe, G., Garcia-March, M. A., Aghion, E., Argun, A., Hong, C. B., Bland, T., Bo, S., Conejero, J. A., Firbas, N., Orts, Ò. G. I., Gentili, A., Huang, Z., Jeon, J., Kabbech, H., Kim, Y., Kowalek, P., Krapf, D., Loch-Olszewska, H., . . . Manzo, C. (2021). Objective comparison of methods to decode anomalous diffusion. Nature Communications, 12(1). <https://doi.org/10.1038/s41467-021-26320-w>

Dmlc. (s. f.). xgboost/doc/parameter.rst at master · dmlc/xgboost. GitHub. <https://github.com/dmlc/xgboost/blob/master/doc/parameter.rst>

APPENDIX

PHENOMENOLOGICAL MODELS

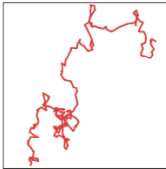
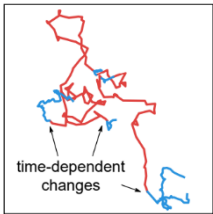
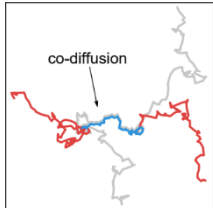
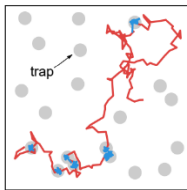
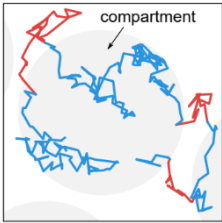
Model	Description	Input	Output
Single State Diffusion 	It models a particle exhibiting a singular diffusive state.	K, α	Tuple containing the trajectories (time, number of trajectories, number of dimensions) and their labels
Multiple State Diffusion 	It explores a particle's transition between diffusive states, with customizable state numbers.	K, α, M M : matrix that contains the probability of jumping from one state to another	
Dimerization 	It examines a set of trajectories all moving within the same environment.	K, α, r, P_b, P_u r : radius P_b : probability of binding when the distance between two particles is less than $2r$. P_u : probability of a dimer of unbinding.	
Immobile Traps 	It accounts for the existence of fixed traps with a radius r that fully immobilize the particles.	K, α, r, P_b, P_u , number of traps, traps positions P_b : probability of getting trapped. P_u : probability of getting released.	
Confinement 	It includes compartments with boundaries that can trap particles within them. We specifically focus on osmotic boundaries, ensuring particles always enter the compartment.	K, α , trans, number of compartments, r , size of the environment trans: probability of the particle exiting the compartment.	

Table 1. Characteristics of the phenomenological models used to create the trajectories

SINGLE-STATE MODEL (SSM):

To generate trajectories for the single-state model, the `single_state` function from the `models_phenom` class in the `andi_dataset` library is employed. This function generates datasets of single-state trajectories in either two or three dimensions using fractional Brownian motion (FBM). First, parameters like the number of trajectories, trajectory length, diffusion coefficient, and anomalous exponent are specified. Internally, it calls the function `_single_state_traj` to create individual trajectories with reflecting boundary conditions if needed. These trajectories are returned with associated labels.

Procedure for `single_state` function:

1. Initialize arrays for positions and labels.
2. For each trajectory (n from 0 to N-1):
 - a. Sample the anomalous exponent α_{traj} and diffusion coefficient D_{traj} using Gaussian distribution (mean and variance specified by `alphas` and `Ds`)
 - b. Generate the trajectory using `_single_state_traj` with these parameters.
 - c. Store the generated positions and labels.

MULTI-STATE MODEL (MSM):

The multi-state model also describes particles undergoing FBM with changing diffusion properties over time. It operates as a Markov model with a fixed number of states S . In this model, each trajectory samples S values of α and S values of K , one for each state. At every time step, there's a probability that the particle transitions between states, determined by a transition matrix M , where $M_{i,j}$ denotes the probability of transitioning from state i to state j at each time step.

The residence time in a given state i can be calculated inversely proportional to the probability of transitioning out of that state: $1/(1-M_{i,i})$

The `multi_state` function is used to generate multiple-state trajectories. This function utilizes various parameters to produce trajectories: `N` for the number of trajectories, `T` for their length, `M` for the transition matrix between diffusive states, `Ds` for diffusion coefficient sampling, and `alphas` for anomalous exponent sampling.

Additionally, parameters such as `gamma_d` and `epsilon_a` control factors between diffusive states. The function returns trajectory data and associated labels, indicating particle positions and their properties over time. These trajectories are generated based on Markovian principles, where particles transition between different diffusive states with defined probabilities. Diverse trajectories reflecting the dynamic nature of particle diffusion are produced by sampling parameters for each state and employing transition matrices, which enables the generation of realistic trajectory data.

Procedure for `multi_state` function:

1. The function first checks the input arguments and ensures that parameters like the transition matrix (M), diffusion coefficients (Ds), and anomalous exponents (`alphas`) are in the correct format (numpy arrays).
2. It samples diffusion coefficients and anomalous exponents for each state based on the provided means and variances.
3. The function iterates over each trajectory (N trajectories in total). For each trajectory
 - a. It samples diffusion parameters (`alphas_traj` and `Ds_traj`) for each state from the distributions determined in the previous step.

- b. It generates the trajectory using the `_multiple_state_traj` function, which simulates the trajectory over time, considering transitions between different diffusive states according to the provided transition matrix (M). Optionally, it can initialize the trajectory with a specific state (`init_state`).
 - c. The generated trajectory and corresponding labels are stored in the `trajs` and `labels` arrays, respectively.
4. Function returns the generated trajectories (`trajs`) and their labels (`labels`).

DIMERIZATION MODEL (DIM):

The dimerization model simulates the transient binding and unbinding of particles, altering their diffusion properties. For generating trajectories, the dimerization function is used, which generates 2D trajectories of particles undergoing stochastic dimerization. Initially, the function sets up parameters such as the number of trajectories (N), trajectory length (T), box size (L), particle radius (r), and binding (P_b) and unbinding (P_u) probabilities. Particle positions are initialized randomly within the box, and their diffusion parameters, including α and K , are sampled from specified distributions.

During each time step, particles move according to their diffusion properties. If two particles come within a distance of $2r$, they have a probability (P_b) of forming a dimer, which then moves with new diffusion parameters specific to the dimeric state. The particles in a dimer have synchronized movements until the dimer breaks with a probability (P_u), after which the particles revert to their monomeric diffusion parameters. Throughout the simulation, the function continuously updates particle positions, handles particle binding and unbinding, and applies boundary conditions.

TRANSIENT-CONFINEMENT MODEL (TCM):

The Transient-Confinement Model simulates particle diffusion within an environment containing multiple non-overlapping circular compartments. Each compartment has a distinct radius r_c and a boundary characterized by transmittance T . Particles can freely enter a compartment upon reaching its boundary from the outside, but once inside, they face a probability T of exiting the compartment upon encountering the boundary again. The diffusion properties inside and outside the compartments differ, with each state defined by unique values of α and K , sampled from predefined distributions.

For each trajectory, it calls `_confinement_traj` to simulate the particle's movement, updating positions and labels at each time step based on whether the particle is inside or outside a compartment. This function initializes particle positions randomly within a box of size L and determines whether each particle starts inside or outside a compartment by calculating the distance from each compartment's center. The function then simulates the motion of particles over time, updating their positions based on their current diffusion state. For particles inside a compartment, it checks if they attempt to exit the compartment upon reaching its boundary, factoring in the transmittance T . If the particle does not exit, it is reflected back into the compartment. If it exits, the function checks if it enters another compartment or remains in the free state.

QUENCHED-TRAP MODEL (QTM):

The Quenched-Trap Model describes particle diffusion within an environment containing immobile traps. During each iteration, the `immobile_traps` function invokes the `__update_bound` function to manage the trapping and untrapping dynamics. This internal function evaluates the proximity of particles to the immobile traps and probabilistically determines whether a particle binds to a trap or unbinds from it. It does this by evaluating the distance between particles and traps, determining if a

particle is within the trapping radius rt . Based on the binding probability P_b and the unbinding probability P_u , particles are bound or unbound accordingly. Following the trapping dynamics, the function updates the positions of the particles, considering both bound and unbound states. Particle motion is adjusted accordingly, reflecting the influence of the traps on their trajectories.

RANDOM FOREST

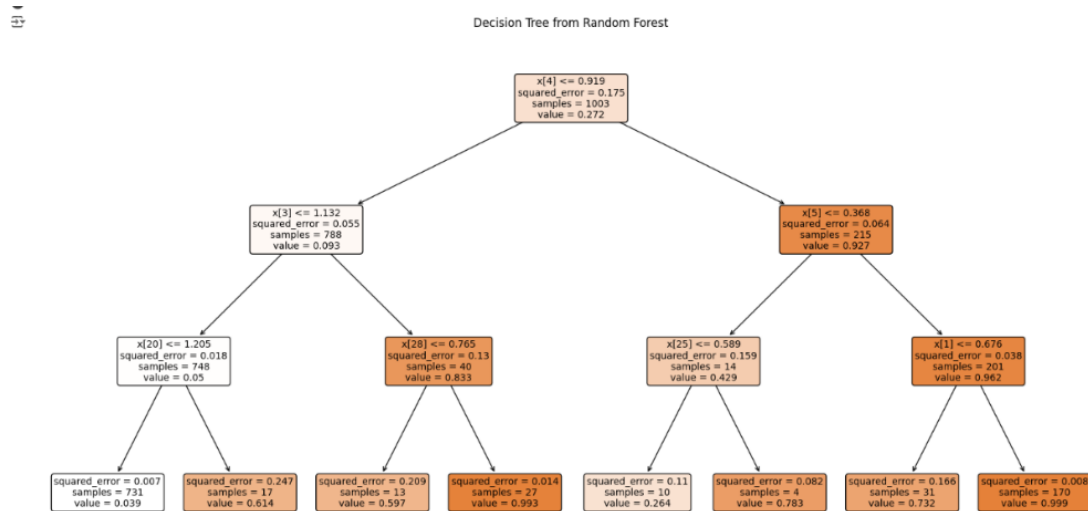


Figure 13. Decision Tree from RF (limited depth for better understanding)

Variable	Importance
2	0.13
0	0.12
8	0.09
1	0.08
3	0.07
4	0.07
6	0.06
9	0.05
12	0.05
5	0.04
11	0.04
10	0.03
7	0.02
15	0.02
16	0.02
19	0.02
13	0.01
14	0.01
17	0.01
18	0.01

20	0.01
21	0.01
22	0.01
24	0.01
25	0.01
27	0.01
28	0.01
23	0.00
26	0.00

Table 2. *Feature Importance RF model*