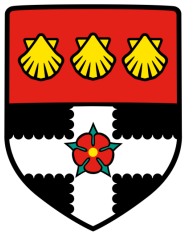


University of Reading

Department of Computer Science

The study of diffusion of nano-particles in polymer and ferrofluids using machine learning

Shreya Venkatesh Desai

Supervisor: Dr. Varun Ojha

A report submitted in partial fulfilment of the requirements of
the University of Reading for the degree of
Master of Science in *Data Science and Advanced Computing*

September 16, 2021

Declaration

I, Shreya Venkatesh Desai, of the Department of Computer Science, University of Reading, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of UoR and public with interest in teaching, learning and research.

Shreya Venkatesh Desai
September 16, 2021

Abstract

Several critical biological and chemical processes are regulated by the molecular diffusion of NPs through a polymer or a carrier fluid. In order to control diffusion, it is imperative to understand the properties of such inhomogeneous networks which influence diffusion. Studies show that various factors such as electrostatic interaction, steric interaction, network porosity and so on affect diffusion. However, the use of machine learning (ML) algorithms in this field seems to be restricted. In this thesis, the molecular diffusion of NP is studied in polymer matrix and ferrofluids is studied using ML. Two data sets are generated by molecular dynamics and Brownian dynamics computer simulations of molecular diffusion of NP in polymer matrix and ferrofluids. Predictive and descriptive ML algorithms are applied on the data sets. In case of polymer, it is found that electrostatic potential (U_0), mesh size (a) and screening length (k) are the important features that affect diffusion. Random forest is the best predictive and classification model to predict and classify diffusion respectively. In case of ferrofluids, the critical feature subsets are unclear. It is observed that extreme gradient boosting is the best predictive model and random forest is the best classification model for ferrofluids.

Acknowledgements

Firstly, I would like to thank the head of the department of computer science at University of Reading for providing me the opportunity to pursue this dissertation. Secondly, I would like to thank my thesis supervisor Dr. Varun Ojha for his immense support, encouragement and guidance throughout the thesis.

I would like to extend my sincere gratitude towards Dr. Patrick Ilg and Dr. Zuowei Wang without whom this thesis would not have been possible. I would like to thank them for helping me to understand the subject matters related to the thesis.

Finally, I would like to express gratitude towards my friends and family member for their support during my thesis.

Contents

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 Aims and Objectives . . . 1
- 1.3 Solution approach . . . 2
- 1.4 Summary of contributions and achievements . . . 2
- 1.5 Organization of the report . . . 2

2 Literature Review **4**
- 2.1 Diffusion . . . 4
  - 2.1.1 Diffusion of nano-particles in polymer . . . 5
  - 2.1.2 Diffusion of ferrofluids . . . 6
- 2.2 Probe diffusion using molecular dynamics simulations . . . 7
- 2.3 Use of Machine learning to study diffusion . . . 9

3 Machine learning algorithms **10**
- 3.1 Descriptive analysis . . . 11
  - 3.1.1 Exploratory data analysis (EDA) . . . 11
  - 3.1.2 Principal component analysis (PCA) . . . 11
  - 3.1.3 Pearson correlation coefficient . . . 11
  - 3.1.4 t-Test . . . 11
  - 3.1.5 Feature selection . . . 12
- 3.2 Predictive analysis . . . 12
  - 3.2.1 Decision tree . . . 12
  - 3.2.2 Random forest . . . 13
  - 3.2.3 Extreme gradient boosting . . . 14
  - 3.2.4 Flexible neural trees . . . 14
  - 3.2.5 Clustering : K-means . . . 15
  - 3.2.6 K-nearest neighbour (KNN) . . . 16
- 3.3 Performance Measures . . . 16
  - 3.3.1 Coefficient of determination (R^2) . . . 16
  - 3.3.2 Precision and recall . . . 17
  - 3.3.3 Receiver operating characteristics (ROC) plot . . . 17

4 Methodology **19**
- 4.1 Computer simulations of diffusion . . . 19
  - 4.1.1 Simulation of diffusion of nano-particles in a cubic array of repelling rods 19
  - 4.1.2 Molecular dynamics simulation of diffusion of magnetic NP in ferrofluids 24
- 4.2 Implementation of machine learning algorithms . . . 26

5 Results and discussion : Diffusion of NP in polymer 28
5.1 Descriptive analysis . . . 28
5.1.1 Exploratory data analysis . . . 28
5.1.2 Pearson correlation coefficient . . . 28
5.1.3 Principal component analysis . . . 30
5.1.4 Feature Selection . . . 31
5.1.5 t-test . . . 31
5.2 Predictive analysis . . . 32
5.2.1 Regression . . . 32
5.2.2 K-means and manual clustering . . . 34
5.2.3 Classification . . . 36

6 Results and Discussion : Diffusion in ferrofluids 39
6.1 Descriptive analysis . . . 39
6.1.1 Exploratory data analysis . . . 39
6.1.2 Pearson correlation coefficient . . . 39
6.1.3 Principal component analysis . . . 40
6.1.4 Feature selection . . . 41
6.1.5 t-Test . . . 41
6.2 Predictive analysis . . . 42
6.2.1 Regression . . . 42
6.2.2 K-means and manual clustering . . . 44
6.2.3 Classification . . . 47

7 Conclusions and Future Work 49

8 Reflection 50

Appendices 55

A An Appendix Chapter 55
A.1 In a nutshell: Molecular Dynamics, Langevin Dynamics, Brownian Dynamics . 55

List of Figures

2.1 (a) Steady-state diffusion across a thin plate. (b) A linear concentration profile for the diffusion situation in (a) (Callister and Rethwisch, 2020). 5
2.2 The connection between the experiment, theory and computer simulation (Allen and Tildesley, 1989). 7
2.3 Schematic of a bead-spring cross-linked network. Each stick connector represents a spring (Zhou and Chen, 2009). 8

3.1 Hierarchy of ML algorithms 10
3.2 Snippet of decision tree created using diffusion data set 13
3.3 Simple structure of FNT 15
3.4 Summary of confusion matrix (Han, Kamber and Pei, 2012). 17
3.5 ROC curves of two classification models M_1 and M_2 (Han, Kamber and Pei, 2012). 18

4.1 Illustration of the regular three-dimensional cubic network (infinitely extended in all three space dimensions). 20
4.2 Schematic of mean-square displacement as function of time (blue). Red dashed lines indicate the short- and long-time diffusive regimes 23
4.3 Left: The mean-square displacement $\langle x^2(t)\rangle, \langle y^2(t)\rangle, \langle z^2(t)\rangle$ as a function of t/τ_{B} for $a^* = U_0^* = 10, \varepsilon^* = 1, k^* = 2$ and ensemble of $N = 100$. The brown line denotes the non-interacting case. Right: Trajectories of four selected particles. 23
4.4 Mean-square displacement $\langle x^2\rangle/(a/2)^2$ scaled with half the mesh size a squared for strongly attractive network, $U_0^* = -20$, $k^* = 2, \varepsilon^* = 1$. 24

5.1 Variation of diffusion (D) with respect to electrostatic potential (U_0), mesh size (a), screening length (K) and strength of LJ repulsion (ϵ) 29
5.2 Variation of diffusion with respect to principal components 30
5.3 Actual vs predicted diffusion 32
5.4 Decision tree 33
5.5 Random forest 33
5.6 Extreme gradient boosting 34
5.7 Comparison of R2 scores of Random Forest (RF), Decision Tree (DT) and XGBoost algorithms based on number of features 34
5.8 Silhouette score for k clusters 35
5.9 Behaviour of k-means with respect to U_0 and D 35
5.10 Histogram of diffusion 36
5.11 Manual clusters plotted on PCA grid 36
5.12 ROC curve of KNN and RF 37

6.1 Variation of particle volume fraction (ϕ), dipolar coupling constant (λ) and Langevin parameter (α) with respect to diffusion (D) 40
6.2 Variation of diffusion with respect to principal components 41
6.3 Actual vs predicted diffusion 43
6.4 Decision Tree 43
6.5 Random forest 44
6.6 Extreme gradient boosting 44
6.7 Comparison of R2 scores of random forest (RF), decision tree (DT) and XG-Boost algorithms based on number of features 45
6.8 Silhouette score for k clusters 45
6.9 Behaviour of k-means with respect to α and D 46
6.10 Histogram of diffusion 46
6.11 Manual clusters plotted on PCA grid 47
6.12 ROC curve of KNN and RF 48

List of Tables

1.1 List of research questions and objectives of the thesis 2

3.1 Parameter settings for decision tree 13
3.2 Parameter settings for random forest 14
3.3 Parameter settings for extreme gradient boosting 14
3.4 Parameter settings for FNT 16

4.1 Parameter settings of BD simulation 21
4.2 Parameter settings of MD simulation 25

5.1 Pearson correlation coefficient of all the features with respect to D 28
5.2 Influence of U_0, a, k, ϵ on the principal components 0 and 1 30
5.3 Feature ranking based on selection rate using flexible neural tree 31
5.4 2-feature subset selection using flexible neural trees 31
5.5 3-feature subset selection using Flexible Neural Trees 31
5.6 t-test on FNT experiment data 32
5.7 Comparison of R^2 scores of n feature subsets where $n = 2, 3, 4$ 33
5.8 Precision (P) and recall (R) scores of random forest 37
5.9 Precision (P) and recall (R) scores of k-nearest neighbour 37

6.1 Pearson correlation coefficient of all the features with respect to D 39
6.2 Influence of ϕ, λ, α on the principal components 0 and 1 41
6.3 Feature ranking based on selection rate using flexible neural tree 41
6.4 2-feature subset selection using flexible neural trees 42
6.5 t-test on FNT experiment data 42
6.6 Comparison of R2 scores with repsect to the number of features 44
6.7 Precision (P) and recall (R) scores of random forest 47
6.8 Precision (P) and recall (R) scores of k-nearest neighbour 47

List of Abbreviations

NP	Nano-particle
NM	Nano-material
MNP	Magnetic nano-particle
ML	Machine learning

Chapter 1

Introduction

1.1 Background

With the advent of nano-technology, it is possible to study and create custom materials with desired mechanical, chemical, electrical, magnetic and other properties giving rise to a new class of biomaterials (Callister and Rethwisch, 2020). Usually, they are synthesized by reinforcing polymers with nano particles (NP) known as nano-composites (Young and Lovell, 2011). This technique is used to engineer materials with amenable features that can respond to external stimulus such as temperature, electrical field, magnetic field and so on. Hence, they are also known as smart materials. They are used to manipulate processes such as molecular diffusion. The central idea of diffusion is transport of mass from one region to another (Callister and Rethwisch, 2020). It which serves critical applications in drug delivery, biomedical science (Kalia, Kango, Kumar, Haldorai, Kumari and Kumar, 2014, Liu, Liu, Cui, Wang, Zhang and Tang, 2020, Meyer and Green, 2015), medicine, industrial (Lopez-Lopez, Durán, Iskakova and Zubarev, 2016), environmental remedies (Kalia, Kango, Kumar, Haldorai, Kumari and Kumar, 2014, Zhu, Wei, Chen, Gu, Rapole, Pallavkar, Ho, Hopper and Guo, 2013) and other industries.

Diffusion is studied either by conducting actual experiments or by performing molecular dynamics computer simulations. However, computer simulations prove to be more convenient to obtain data and perform analysis. Inspite of several studies conducted in this area using computer simulations, the capabilities of machine learning are rarely utilised. It is necessary to understand the nature of the interaction between nano-particles and the polymer matrix or carrier fluid as well as its effects on molecular diffusion. This thesis proposes to fill these existing gaps by producing descriptive and predictive models ML to gain an in-depth understanding of the factors affecting diffusion. This will guide the scientists towards a focused approach to engineer biomaterials with desired characteristics.

1.2 Aims and Objectives

The aim of this thesis is to study diffusion in inhomogeneous networks consisting of NPs, MNPs and polymers by using ML. ML will help in identifying how the behaviour of NPs and MNPs as well as the properties of polymer affect the diffusion process. This understanding can be extrapolated to engineer effective biomaterials. The aim is broken down into several objectives to answer the prominent research questions as shown in the Table 1.1.

Table 1.1: List of research questions and objectives of the thesis

No.	Research Questions	Objectives
1	How diffusion depends on the factors?	Identify comparative strength of influence of the factors on tracer particle diffusion
2	Can we identify the critical factors that influence diffusion?	To conduct focused study on the critical factors
3	Can we predict diffusion based on the values of the factors?	To avoid simulations and save time
4	Can we categorise diffusion?	To detect any similarities or patterns in the data and label them
5	Can we classify diffusion based on the categories?	To understand the diversity in diffusion with respect to the feature values, similarities and differences

1.3 Solution approach

The methodology adopted in this thesis begins with data collection phase using computer simulations. Simulations of self diffusion of MNP in ferrofluids generate data set containing mesh size (a), electrostatic potential (U_0), screening length (k), strength of LJ repulsion (ϵ) and diffusion (D). The data set of self diffusion of NP in polymer matrix consists of particle volume fraction (ϕ), dipolar coupling constant (λ), Langevin parameter (α), diffusion (D), average cluster size (savg1) and magnetization (M). Next, we apply predictive and descriptive machine learning algorithms to both the data sets to answer the research questions in Table 1.1.

1.4 Summary of contributions and achievements

Predictive (regression and classification) models with high accuracy are obtained to predict diffusion based on the feature values. Random forest is the best predictive and classification model to predict and classify diffusion respectively. It is observed that extreme gradient boosting is the best predictive model and random forest is the best classification model for ferrofluids. In case of polymer, it is found that electrostatic potential (U_0), mesh size (a) and screening length (k) are the important features that affect diffusion. In case of ferrofluids, the critical feature subsets cannot be clearly identified.

1.5 Organization of the report

The thesis is outlined as follows:

- **Chapter 2 - Literature Review**
  In this section, the study provides a basic context of nano particles, nano materials and magnetic nano particles. Using it as a basis, it further explains the central idea of diffusion and the studies present in the literature. It mentions the factors influencing diffusion, process of computer simulations and exposes the gaps related to use of ML to study diffusion in the existing literature.

- **Chapter 3 - Machine learning techniques**
  A brief description of machine learning algorithms is given along with their advantages and working.

- **Chapter 4 - Methodology**
  This chapter explains the detailed process of data collection using computer simulations along with the application of ML techniques on the data sets.

- **Chapter 5 - Results and discussion : Diffusion ofNP in polymer**
  The results of ML are analysed and their implications are discussed in depth.

- **Chapter 6 - Results and Discussion : Diffusion in ferrofluids**
  The results of ML are analysed and their implications are discussed in depth.

- **Chapter 7 - Conclusion and future work**
  The results are summarised and future work is suggested.

- **Chapter 8 - Reflection**
  This section describes individual learning experience gained throughout the research and implementation of thesis.

Chapter 2

Literature Review

With the advent of technological advancements such as microscopes, the scientists are able to study the complex structures on an atomic (microscopic) level (Callister and Rethwisch, 2020). Observation of physics and chemistry of atoms (also known as the "building blocks of matter"), has lead to the discovery of nano-particles (NP) whose size is of the order of nanometer ($10^{-9}m$) (Callister and Rethwisch, 2020). Nano-materials (NM) are materials whose internal structure is made up of several NPs. Further investigations reveal that NMs of type metals, semi-conductors, ceramics, polymer and composites exhibit drastically different physical and chemical properties at an atomic level as opposed to the macroscopic level (Callister and Rethwisch, 2020). Using NMs, it is feasible to design and create new materials with custom mechanical, electrical, magnetic and other properties, thus, giving rise to the age of "materials by design" (Callister and Rethwisch, 2020). Exploiting this concept, materials with tractable features are engineered that can be manipulated using an external stimulus such as pH, temperature, chemical, electrostatic field, light, magnetic field, shear-sensitive and so on (Wang, Li, Ouyang and Karniadakis, 2020). These materials are used to regulate the process of molecular diffusion which find several critical applications. Thus, it is imperative to study the process of diffusion to engineer smart materials.

2.1 Diffusion

Historically, the term diffusion has been extensively used in physics, biology, chemistry, sociology, economics and finance. However, this literature review focuses on the definition and applications of molecular diffusion corresponding to physics, chemistry and biology. To begin with, the process of diffusion is a natural phenomenon and can be observed in everyday life. There are many common place examples of diffusion in solid, liquid and gas. One such example is that of a perfume bottle. When it is opened, the fragrance spreads in the air since the atoms of the perfume liquid diffuse in the air. Intuitively, this leads to the basic definition of diffusion as transport of mass by atomic movement (Callister and Rethwisch, 2020).

Diffusion was first mathematically described by Fick's Laws derived by Adolf Fick in 1855 (Callister and Rethwisch, 2020). Considering a linear concentration profile of diffusion of gaseous species (Figure 2.1), the concentration gradient is given by Equation 2.1 (Callister and Rethwisch, 2020).

$$concentration\,gradient = \frac{\Delta C}{\Delta x} = \frac{C_A - C_B}{x_A - x_B} \tag{2.1}$$

Fick's first law for steady-state diffusion in a direction x states that the flux (J) is proportional to the concentration gradient given by the Equation 2.2 followed by the second law for non-

steady state diffusion (Callister and Rethwisch, 2020) .

$$J = -D\frac{dC}{dx} \tag{2.2}$$

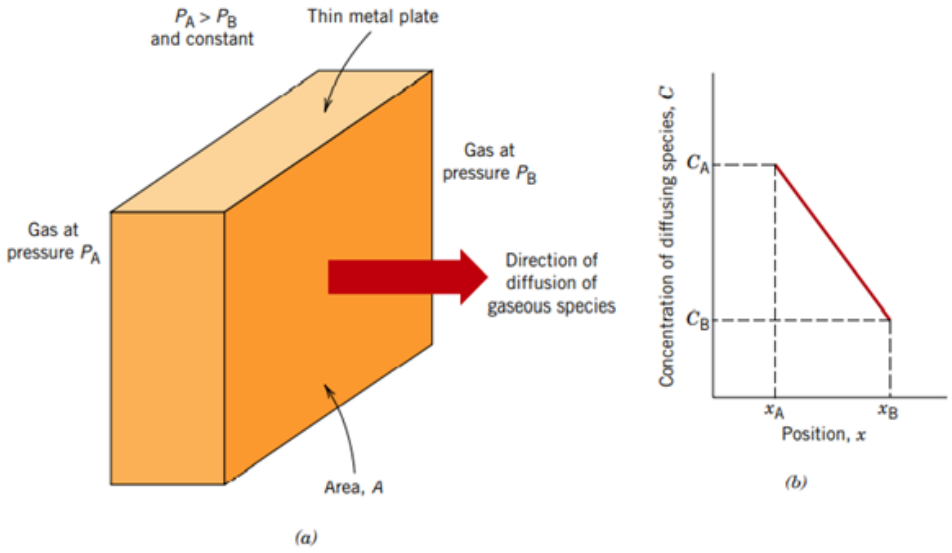


Figure 2.1: (a) Steady-state diffusion across a thin plate. (b) A linear concentration profile for the diffusion situation in (a) (Callister and Rethwisch, 2020).

On the other hand, Brown (1828) defines diffusion as a random walk of particles suspended in fluid, later developed by Einstein (1905) as Brownian motion. It is given by the first-order differential equation (Equation 2.3) where, $F_i = -\nabla_i U$ are potential forces, ξ is the friction coefficient, F_i^B represents Gaussian white noise and r_i is the displacement of the NP i (Einstein, 1905). For spherical particles, $\xi = 3\pi\eta\sigma$ where η is the viscosity of the solvent and σ is the hydrodynamic diameter (Einstein, 1905).

$$\frac{d}{dt}r_i = \frac{1}{\xi}F_i + \frac{1}{\xi}F_i^B \tag{2.3}$$

2.1.1 Diffusion of nano-particles in polymer

Polymer is a substance composed of molecules that have lengthy sequences of one or more species of atoms connected to each other covalently (Young and Lovell, 2011). They possess peculiar chemical and physical properties such as high impact tolerance and elasticity as compared to other metals or semi-conductors (Pethrick, Amornsakchai and North, 2014). As a result, there are numerous applications of polymers alone as well as their enhanced variations. Generally, polymers are reinforced by nanoparticles forming nanocomposites (Young and Lovell, 2011). Due to miniscule size of the nano particles, the surface-to-volume ratio of nanocomposites increases. Thus, they offer better stress support and are strong as compared to a single polymer matrix (Young and Lovell, 2011). Therefore, extensive research is conducted to study the properties of polymers and utilize them.

The process of diffusion of nano-particles in base polymer or polymer nano-composites finds critical applications in several areas such as chemical engineering, biological processes,

pharmaceutical industries and so on. One of the important applications of this process is in controlled drug delivery (Zhao, Kim, Cezar, Huebsch, Lee and Mooney, 2011). When a drug is administered orally or through injection, the concentration of the drug in the body begins to surge to a high value and then decreases in an exponential manner. It is necessary to maintain an initial concentration over a reasonable time period to attain the desired effect. The solution to this problem is to incorporate the drug into a polymer from which it can diffuse inside the body at a planned rate. A study (Wang, Li, Ouyang and Karniadakis, 2020) exploits this mechanism to control the diffusion of NP (drug) in thermo-responsive (temperature sensitive) hydrogel polymer network by changing the temperature. Thus, the drug release kinetics (diffusion) can be controlled to design smart drug delivery system. Another useful application (Lieleg, Baumgärtel and Bausch, 2009) of this mechanism is to use partially swollen hydrogels formed from cross- linked hydrophilic polymers (Pethrick, Amornsakchai and North, 2014) as film to cover injuries such as burns. The film allows controlled transit of water along with antibiotic substances while blocking the bacteria to prevent infections. Similarly, hydrophilic polymers are used in the treatment of waste water by allowing water molecules to diffuse freely while blocking the large harmful bacteria. In industries, it is common to form bicomponent gels by mixing 2 immiscible polymer gels using nano particles as catalyst (Chen and Yong, 2019). The NP increase the entropy and make the reaction more spontaneous facilitating their mixing (Chen and Yong, 2019). When the NPs interact with the polymer matrix (polymer gel), the diffusion of NP is affected by many factors such as steric effects (Fatin-Rouge, Starchev and Buffle, 2004), network porosity, flexibility, degree of cross linking, temperature, electrostatic interaction (Zhang, Hansing, Netz and Derouchey, 2015) and so on (Zhou and Chen, 2009).

2.1.2 Diffusion of ferrofluids

Magnetism is a well-known phenomenon where certain materials possess an attractive or repulsive force or influence on other materials (Callister and Rethwisch, 2020). Some substances such as lodestone, naturally possess magnetism whereas, in case of other substances such as iron which are sensitive to magnetic field, magnetism can be induced by using an external magnetic field (Callister and Rethwisch, 2020). With the advent of nano-technology, extensive research is being conducted in developing, studying and adopting magnetic nano-materials and their corresponding magnetic polymer nanocomposites. Due to their sensitivity towards magnetic field, their physical, mechanical and chemical properties can be remotely controlled by externally manipulating the strength of the magnetic field. This special feature gives rise to numerous biomedical (Kalia, Kango, Kumar, Haldorai, Kumari and Kumar, 2014, Liu, Liu, Cui, Wang, Zhang and Tang, 2020, Meyer and Green, 2015), industrial (Lopez-Lopez, Durán, Iskakova and Zubarev, 2016), optical (Li, Meng Lin, Toprak, Kim and Muhammed, 2010) and environmental (Kalia, Kango, Kumar, Haldorai, Kumari and Kumar, 2014, Zhu, Wei, Chen, Gu, Rapole, Pallavkar, Ho, Hopper and Guo, 2013) applications.

Ferrofluids are magnetic nano-materials who have gained a lot of attention in the recent years because their physical properties can be controller by an external magnetic field. They are colloidal liquid in which magnetic nano particles are suspended (Ilg and Kröger, 2005). Though their gradient diffusion in ferrofluids is studied experimentally and theoretically, self-diffusion or tracer-diffusion in ferrofluids has rarely been explored (Ilg and Kröger, 2005). Some studies (Ilg and Kröger, 2005, Wang, Holm and Müller, 2002) show that in the presence of an external magnetic field, the anisotropic self diffusion depends on the strength of the dipolar interaction between the particles, concentration of the number of particles, susceptibility and the magnetic field strength. It also found that the formation of clusters amongst the magnetic nano particles enhances the magnetization in the ferrofluids.

2.2 Probe diffusion using molecular dynamics simulations

Probe diffusion also called as microrheology in the literature, is used to study the rheology of polymer solutions, colloidal gels and hydrogels (Zhou and Chen, 2009). At the very beginning, scientists experimented with physical objects such as gelatine balls or hard spheres to represent the molecular structure to study the rheology of liquids (Allen and Tildesley, 1989). However, such methods are extremely time consuming especially when the number of particles increase and each of their interactions need to be calculated manually. Allen and Tildesley (1989) propose computer simulations which are fast, reliable and convenient to test the existing approximation methods and shed light on the new approaches. Further, they illustrate the connection between the experiment, theory and computer simulation as shown in Figure 2.2. They propose that the results of simulations can be compared with the results derived from real experiments and theoretic predictions ensuring the testing of the underlying model to perform necessary corrections in the model.

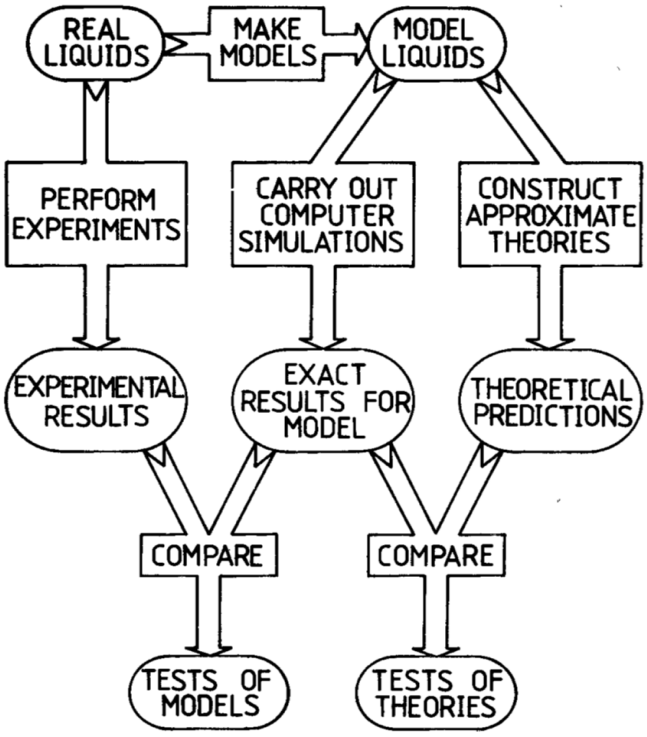


Figure 2.2: The connection between the experiment, theory and computer simulation (Allen and Tildesley, 1989).

Probe diffusion requires a structural and mathematical framework for performing computations. The structural framework defines the molecular arrangement by specifying the number of dimensions, type of lattice, degree of cross-linking of the atoms and molecules, charged or uncharged matrix and so on. Different 3-dimensional lattice models such as coarse-grained bead-spring (Figure 2.3) and bead-rod are used to represent the polymer architectures (Zhou and Chen, 2009, Cruz, Chinesta and Régnier, 2012). Usually, a cubic lattice is constructed where each bead is positioned on a cross-link point to 6 other adjacent points with the help of hookean springs (Cruz, Chinesta and Régnier, 2012). The degree of cross-linking can be manipulated as per research requirement (Cruz, Chinesta and Régnier, 2012).

Computer simulations use statistical mechanics as mathematical framework to compute

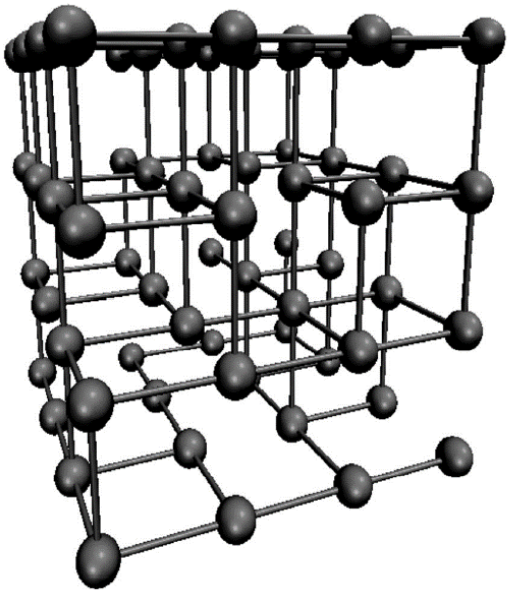


Figure 2.3: Schematic of a bead-spring cross-linked network. Each stick connector represents a spring (Zhou and Chen, 2009).

exact results. Statistical mechanics establishes a theoretical link between the microscopic and macroscopic descriptions (Allen and Tildesley, 1989). Microscopic description of matter involves masses of atoms, interaction between the atoms, molecular geometry, positions and velocities of molecules and atoms. On the other hand, macroscopic description of matter uses observables like diffusion, temperature, pressure, density and so on. Thus, statistical mechanics encapsulates the molecular dynamics at a finer scale. In case of diffusion, Fick's laws provide a macroscopic definition of diffusion resonating with thermodynamics whereas Brown (1828) presents a microscopic definition of diffusion. Molecular dynamics (MD), Langevin dynamics (LD), Brownian dynamics (BD) and Monte-Carlo (MC) simulations are some of the common techniques of statistical mechanics used to model diffusion. Chen and Kim (2004) review the methods, their comparative strengths and weaknesses along with their applications. Brownian equation is derived from Langevin equation which is in-turn derived from the basic definition of molecular dynamics. The details of the mathematical derivations are provided in Appendix A. BD addresses the problem of timescale separation when one form of motion in a system is much faster than the other (Allen and Tildesley, 1989). Shorter times-steps and long runs of simulations are required to allow the progression of the slower motions (Allen and Tildesley, 1989). For example, while simulating a large number of molecules in a solvent, the fast motions of the solvent molecules are not of interest and need to be ignored in order to make the simulations less expensive (Allen and Tildesley, 1989). Langevin equation solves this problem by omitting them from the simulation and representing their effects on the solute particles as a combination of random forces and frictions (Allen and Tildesley, 1989). Ensemble is a collection of a large number of identical systems (M) evolving in time under the same macroscopic conditions but different microscopic initial conditions (Allen and Tildesley, 1989). Macroscopic properties (observables) are then calculated as weighted averages. Molecular methods are similar to experiments where weighted averages are computed (Allen and Tildesley, 1989). This process of ensemble average is a part of the statistical mechanics which is used to calculate the diffusion (macroscopic property) of NPs in polymer. Nontheless, Brownian dynamics and Langevin dynamics computer simulations are

some of the popular methods employed to study diffusion (Zhou and Chen, 2009, Ryzhkov, Melenev, Holm and Raikher, 2015, Megariotis, Vogiatzis, Schneider, Müller and Theodorou, 2016, Zhang, Hansing, Netz and Derouchey, 2015, Ilg and Kröger, 2005, Wang, Holm and Müller, 2002).

2.3 Use of Machine learning to study diffusion

Substantial research (Zhou and Chen, 2009, Ryzhkov, Melenev, Holm and Raikher, 2015, Megariotis, Vogiatzis, Schneider, Müller and Theodorou, 2016, Zhang, Hansing, Netz and Derouchey, 2015, Ilg and Kröger, 2005, Wang, Holm and Müller, 2002) have been conducted to observe factors that influence diffusion by performing MD/BD computer simulations of NP and MNP. However, it is difficult to calculate the statistically relevant translational diffusion coefficients on shorter time scales. Therefore, longer simulations are required to obtain the necessary values/information which are critical to the study. Furthermore, these studies conduct computer-based simulations or actual experiments which is time consuming and computationally costly to perform repeatedly for many polymers.

On the other hand, several studies use machine learning to study and characterize diffusion. Chen and Yong (2019) apply artificial neural network supervised machine learning (ML) algorithm to predict the polymer phase of bicomponent hydrogels, by training model with some initial experimental data containing radii and volume of Janus nanoparticles which facilitate the formation of hydrogels. They use the k - nearest neighbours (k-NN) unsupervised (ML) algorithm to classify the predicted phase. Muñoz-Gil, Garcia-March, Manzo, Martín-Guerrero and Lewenstein (2019), Granik, Weiss, Nehme, Levin, Chein, Perlson, Roichman and Shechtman (2019) use artificial neural network (ANN) and deep learning respectively to detect the anomalous diffusion based on mean square displacement (MSD).

Inspite of adopting machine learning techniques to study diffusion, they fail to build predictive models to predict diffusion based on the values of parameters affecting it. In order to effectively control diffusion, it is required to understand the magnitude of their influence on the diffusion process and isolate the most important parameters. Therefore, this thesis proposes to fill the existing gaps and leverage the capabilities of ML by producing predictive models with high accuracy. The ML models can be reused in the future to predict diffusion without performing any computer simulation or actual experiments thus expediting research. Using feature analysis, we can identify the critical parameters that influence diffusion. This will further guide the researchers towards a focused study to design biomaterials which specifically incorporate the important parameters to regulate diffusion. Moreover, ML clustering algorithms can be used to label similar data points with respect to diffusion. The classifier models can be trained to classify the labelled data points into meaningful categories. Exploratory data analysis can help to visualise the dependence of diffusion on the given parameters. Thus, this thesis aims to employ the capabilities of ML to extract a detailed schematic of the diffusion process which will serve as a guide towards future study.

Chapter 3

Machine learning algorithms

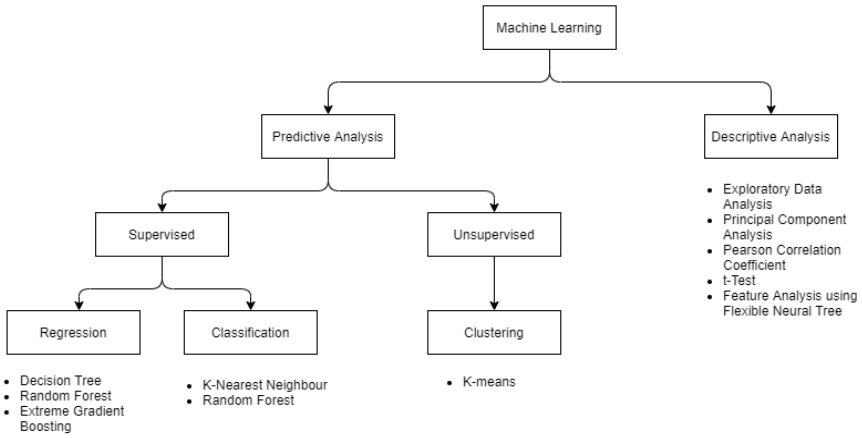


Figure 3.1: Hierarchy of ML algorithms

Machine Learning (ML) is a powerful tool to extract patterns from the given data. Given a set of inputs $x = x_1, x_2, x_3 \dots x_N$ and their corresponding set of target values $\hat{y} = y_1, y_2, y_3 \dots y_N$, ML establishes a mathematical relationship between them in the form of $\hat{y} = f(x)$ (Bishop, 2006). Usually, the data set is divided into the training set and test set. Training set is used to train the models to identify the relationship between the input and the target values (Bishop, 2006). The trained model is validated using the test set and performance measures. Referring to Figure 3.1, in supervised learning past data is used to train the model (Bishop, 2006). On the contrary, unsupervised learning (clustering) tries to find patterns without any historical data (Bishop, 2006). When the target values $\hat{y}$ are continuous, it is treated as a regression problem else if they are discrete (labels) then it becomes a classification problem. Thus, these methods collectively form the predictive analysis. On the other hand, descriptive analysis tries to find patterns using correlations, principal component analysis, feature analysis, exploratory data analysis and so on (Tan, Steinbach and Kumar, 2016). It is primal in nature and sums up the concealed data associations (Tan, Steinbach and Kumar, 2016).

3.1 Descriptive analysis

3.1.1 Exploratory data analysis (EDA)

Cakmak and Cuhadaroglu (2018) describe EDA as one of the critical processes before applying machine learning algorithms and other statistical analysis. It helps to visualize the data without any underlying assumption and identify patterns, data characteristics (Cakmak and Cuhadaroglu, 2018). It paves the way for formulating an appropriate premise and provides an initial headstart to further analysis (Cakmak and Cuhadaroglu, 2018). The process includes plotting coloured histograms, pairplots, scatter plots to understand the general trends and relationships between the variables. (McKinney, 2018). They can be further quantified by computing correlation, averages, variances, principal component analysis. Adopting this process, one can understand the data at a higher level of abstraction.

3.1.2 Principal component analysis (PCA)

PCA was first introduced by Karhunen (1946) and Loève (1948). It decomposes the input data set into most and least significant features affecting the target variable called as principal components (Tan, Steinbach and Kumar, 2016). It computes the most important features so that ML algorithms can be applied to the reduced data set. A reduction in the feature space can significantly impact the performance of the ML models (Tan, Steinbach and Kumar, 2016). PCA helps to identify the set of features which account for the highest variation in the data set (Tan, Steinbach and Kumar, 2016). Even though discarding few features can lead to loss of information, the actual data can be reconstructed with high accuracy using the strong features (Tan, Steinbach and Kumar, 2016).

3.1.3 Pearson correlation coefficient

Correlation mathematically quantifies the magnitude and direction of the relationship between two variables. Pearson correlation coefficient was first introduced by Pearson (1895). The coefficient $r_{A,B}$ is given by Equation 3.1. Here, n is the number of tuples, a_i and b_i are the respective values of A and B in tuple i , $\bar{A}$ and $\bar{B}$ are the respective mean values of A and B , σ_A and σ_B are the respective standard deviations of A and B and $\sum a_i b_i$ is the sum of the AB cross-product (Han, Kamber and Pei, 2012, Chapter 3).

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B} \tag{3.1}$$

The value of $r_{A,B}$ ranges from $+1$ (positive correlation) to -1 (negative correlation). If two variables are positively correlated then the increase in value of one implies an increase in value of the other (Han, Kamber and Pei, 2012, Chapter 3). If two variables are negatively correlated then the decrease of one variable implies an increase in the other (Han, Kamber and Pei, 2012, Chapter 3). 0 means no correlation between the two variables. The correlation can be visualized by using a simple scatter-plot. All the related theory can be found in (Han, Kamber and Pei, 2012, Chapter 3).

3.1.4 t-Test

The t-test was first introduced by William Sealy Gosset as a Student (Student, 1908). It is used to test the null hypothesis that the means of two groups A and B are equal $H_0 : \mu_A = \mu_B$ (Iversen and Gergen, 1997). Using distribution theory results for two independent normal variables, the two-sample t test statistic is given by Equation 3.2 which follows the student's

distribution with n_A+n_B-2 degrees of freedom (Iversen and Gergen, 1997). Here n_A and n_B are the size of the two samples A and B respectively. n is the average size of the two samples, s is the standard deviation, $\bar{x_A}$ and $\bar{x_B}$ are the means of the two samples A and B respectively. If the p-value (probability) calculated is less than 0.05 then there exists statistically significant evidence to reject the null hypothesis H_0, otherwise H_0 is true (Iversen and Gergen, 1997). A large t score indicates that A and B are different and similar otherwise.

$$t = \frac{\sqrt{n}(\bar{x_A} - \bar{x_B})}{s(\frac{1}{n_A} + \frac{1}{n_B})} \tag{3.2}$$

3.1.5 Feature selection

Feature selection is used to evaluate the importance of features in the dataset. It reveals the data features which critically impact the prediction of a target value by a machine learning model (Tan, Steinbach and Kumar, 2016). These features are used to train the machine learning models and their performance is compared to that of models trained with all the data features (Tan, Steinbach and Kumar, 2016). In this thesis, FNT is used perform feature subset selection. Using the parameter settings in Table 3.4 and the FNT tool provided by Ojha (2016), 10-Fold cross validation is performed on the data sets by randomly splitting the datasets into training and test set into ten parts. This experiment is repeated 30 times. In each iteration, selection rates of all the features and all possible combinations of feature subsets are calculated.

3.2 Predictive analysis

3.2.1 Decision tree

The classification and regression trees (CART) was first introduced by Gordon, Breiman, Friedman, Olshen and Stone (1984). It is a very popular algorithm which recursively divides the given input feature space into regions based on different conditions (Bishop, 2006, Chapter 14). A condition is derived based on the splitting attribute which is in turn determined by using different measures such as gini index, information gain and so on (Bishop, 2006, Chapter 14). A label (in case of classification) or a number (in case of regression) is assigned to each constructed region using the input features. In order to predict a target value, the constructed tree is traversed from root node to the appropriate region (leaf node) based on its feature values (Bishop, 2006, Chapter 14). Once the region of the target (predictive) value is identified, its value is predicted by averaging the input data points belonging to that identified region (Bishop, 2006, Chapter 14). The decision tree falls under the umbrella of interpretable machine learning algorithms (Bishop, 2006, Chapter 14) since the reason behind the predicted value can be easily understood by looking at the generated tree as shown in Figure 3.2.

According to Figure 3.2, Uzero and mesh denote the splitting attributes U_0 and a respectively. Each rectangle refers to a region and the samples indicate the number of data samples belonging to that particular region based on the condition. The value is the average diffusion of the data samples in the region. The values of hyper-parameters are adjusted to create a tree with high accuracy in a process called the hyper-parameter tuning. Some of the hyper-parameters are minimum number of samples required to split, maximum depth of the decision tree, maximum leaf nodes and so on. Further details on decision tree can be found in Bishop (2006, Chapter 14). The hyper-parameters used in this thesis are given in Table 3.1.

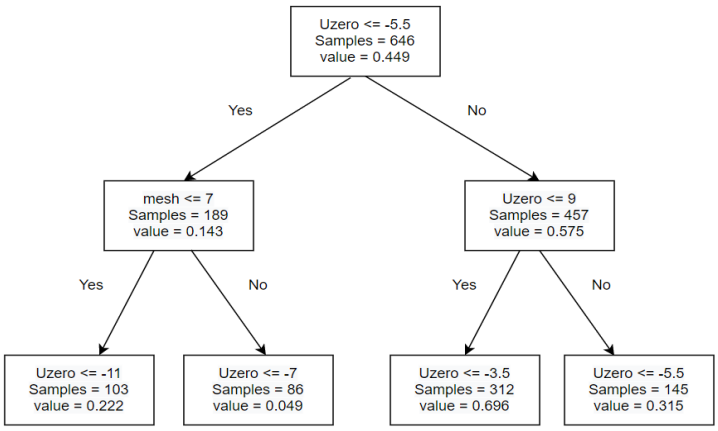


Figure 3.2: Snippet of decision tree created using diffusion data set

Table 3.1: Parameter settings for decision tree

Parameter	**Significance**	**Values**
max_depth	Maximum tree depth	10
min_samples_leaf	The minimum number of samples required to be at a leaf node	1
min_samples_split	The minimum number of samples required to split an internal node	2

3.2.2 Random forest

It was first introduced by Gordon, Breiman, Friedman, Olshen and Stone (1984). RF is an ensemble classifier which consists of many decision trees (DT) (Bishop, 2006, Uddin, Khan, Hossain and Moni, 2019). It is analogous to several trees present in a forest. Random forests is defined as an ensemble of decision tree classifiers $\{h(x, \Theta_k), k = 1, ...\}$ where Θ_k is a random vector and each tree votes for the popular class at input x (Pavlov, 2019). During training, a random subset of the training data is selected. Random features of the subset are selected based on node impurity. Then a node in the decision tree is created by using the feature which gives the maximum classification. This process is repeated for n estimators (Kelkar and Bakal, 2020). The performance of RF depends on the hyper-parameters such as the number of decision trees (n estimators), maximum depth of each decision tree in a forest (max depth), minimum number of tuples required to split each node (min samples split) and minimum number of samples for each leaf node (min samples leaf) (Kelkar and Bakal, 2020). It also depends the number of input features, feature selection, sample size and many other factors as pointed out by (Kelkar and Bakal, 2020). In order to classify a new input tuple, it is passed down to each decision tree (Pavlov, 2019). Each decision tree considers different features of the input tuple and predicts a class label (Pavlov, 2019). Similarly, this process is performed by all the other DTs in the ensemble. Finally, RF assigns the class label with the highest votes from all the DTs to the input tuple (Pavlov, 2019). Since it considers outcomes from several DTs it can be used to reduce the amount of caused variance of a single DT (Pavlov, 2019). Over-fitting to the training data is one of the disadvantages of RF (Pavlov, 2019). The hyper-parameters used in this thesis are given in Table 3.2.

Table 3.2: Parameter settings for random forest

Parameter	Significance	Values
min_samples_leaf	The minimum number of samples required to be at a leaf node	1
min_samples_split	The minimum number of samples required to split an internal node	2
n_estimators	The number of trees in the forest	100

3.2.3 Extreme gradient boosting

Boosting is a very popular type of ensemble methods which use multiple "base" ML models to predict the target variable. It was first introduced as "Adaptive Boosting" by Schapire (2013). The prediction is performed by computing the average result of a set of base classifiers such as decision tree Schapire (2013). Each base classifier is trained using the weighted form of the data set. The weight is decided based on the performance of the previous classifiers Schapire (2013). If the previous classifiers made an error in prediction for some data points, then their weightage is increased in the subsequent classifiers to ensure that they learn data points well and the error is minimized Schapire (2013). Hence, they are known to yield better results than a single base ML model by significantly reducing the overall variance of the model Schapire (2013). Basically, for a given unseen data point, a set of classifiers are built and the average of their predicted values is considered as the value of target variable for the given input data point Schapire (2013). Extreme gradient boosting (EGB) was first introduced by Chen and Guestrin (2016) and is a variant of the gradient boosting (Friedman, 2001) using decision trees. EGB uses a regularization function to retain the generalization ability of the model and avoids over-fitting to the training data set (Friedman, 2001). The hyper parameters such as γ (minimum loss allowed for split), regularization weights (α and λ and so are used to tune the performance of EGB for a given data set (Friedman, 2001). The hyper-parameters used in this thesis are given in Table 3.3.

Table 3.3: Parameter settings for extreme gradient boosting

Parameter	Significance	Values
base_score	The initial prediction score of all instances	0.5
learning_rate	Boosting learning rate	0.1
max_depth	Maximum tree depth for base learners	3
gamma	Minimum loss reduction required for splitting leaf node of a tree	0
n_estimators	Number of boosting rounds	100

3.2.4 Flexible neural trees

Flexible neural tree (FNT) was first introduced by Chen, Yang, Dong and Abraham (2005) used for time series forecasting. It is one of the developments of the artificial neural network (ANN) where a NN takes the form a tree-like structure. All the nodes in a single layer of ANN are use the same activation function, contrary to FNT where it can be different. FNT consists of 3 components: internal nodes, branches and leaf nodes (Ojha, Schiano, Wu, Snasel and Abraham, 2018). The branches are similar to the weighted connections in an ANN. The

internal nodes behave as computational nodes which are nothing but activation functions and the leaf nodes, also referred to as terminal nodes are inputs (Ojha, Schiano, Wu, Snasel and Abraham, 2018). The root node of the tree represents the predicted output of the model which in our case is diffusion. The leaf nodes in FNT indicate the selected features: U_0, a, ϵ and k. Figure 3.3 illustrates the structure of a FNT generated from the input data set using the tool provided by (Ojha, 2016). The root node 2 represents the output diffusion value. Terminal nodes 0 and 1 represent features a and U_0 are selected. The parameter settings are given in Table 3.4. The tree structure is optimized using genetic programming (Ojha, Abraham and Snášel, 2017) which is a evolutionary algorithm based on population (Ojha, Schiano, Wu, Snasel and Abraham, 2018). The parameter optimization is performed using differential evolution (Ojha, Schiano, Wu, Snasel and Abraham, 2018). Further explanation

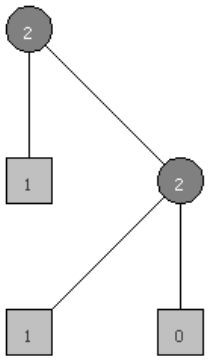


Figure 3.3: Simple structure of FNT

on elitism, mutation, crossover and tournament can be found in Goldberg (1989).

3.2.5 Clustering : K-means

K-means (Tan, Steinbach and Kumar, 2016, Chapter 5) is an unsupervised algorithm which was first introduced by Lloyd (1982) and MacQueen et al. (1967). It is a very popular algorithm to cluster the data sets and find meaningful relationships within the given data. Tan, Steinbach and Kumar (2016, Chapter 5) describe the k-means algorithm in detail. k-means randomly selects k data points as centroids where k is the number of clusters. It assigns all other data points to the clusters based on their minimum eucledian distance from the centroids. This is performed for multiple iterations until there is no or minimal difference in the centroid values. Therefore, each group of similar data points are assigned to a single cluster. This form of labelling helps identify the hidden patterns in the data providing useful information. K needs to be selected very carefully so as to obtain good quality of clusters. Silhouette coefficient (Tan, Steinbach and Kumar, 2016, Chapter 5) is one of the methods for selecting optimal K. The lower the coefficient value, higher the quality of clusters (Tan, Steinbach and Kumar, 2016, Chapter 5).

Table 3.4: Parameter settings for FNT

Parameter	Significance	Values
Max tree depth	Maximum height of the tree	5
Max tree arity	Maximum children nodes can have	4
Function node type	Type of activation function	Gaussian
Genetic programming population	Population size used in optimization of FNT	20
Elitism	Number of best individuals of current population carried forward to next generation	2
Mutation rate	Mutation Frequency	0.2
Crossover rate	Crossover frequency	0.8
Tournament size	Number of individuals part of next generation	2
Meta heuristic algorithm	Parameter optimization method	Differential Evolution
Cross-validation	Type of model validation method	k-fold
Number of Folds	Number of folds of cross validation	10

3.2.6 K-nearest neighbour (KNN)

KNN (Tan, Steinbach and Kumar, 2016) is a non-parametric method which was first introduced by Cover and Hart (1967). According to Tan, Steinbach and Kumar (2016), it is a type of supervised learning used to classify the data points into relevant labels. Given a new data point, its distance from each training data point is calculated. K nearest neighbours are selected based on the minimum distance. The label of the majority of the neighbours is assigned to the new data point. It is a very simple algorithm but provides high performance. The optimal K can be obtained by performing some trials on the data set. KNN is also sensitive to the choice of the distance measure used to calculate the nearest neighbours. In this thesis, $k = 5$ is used to configure the model.

3.3 Performance Measures

The performance indices are used to compute and compare the performances of the ML models. They serve as quantifiable measure of the performance. In this section, the performance indices related to regression and classification ML algorithms are discussed.

3.3.1 Coefficient of determination (R^2)

It was first introduced by Sewall (1921). It is one of the most common methods to determine the accuracy of the predictive ML models for regression problem. It is given by Equation 3.3.

$$R^2 = \left(\frac{\sum_{i=1}^{N} (o_i - \bar{o}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N} (o_i - \bar{o})^2} \sqrt{\sum_{i=1}^{N} (y_i - \bar{y})^2}} \right)^2 \tag{3.3}$$

Where o_i is the predicted output, $\bar{o}$ is the mean of predicted output, y_i is the actual target value and $\bar{y}$ is the mean of the actual target value in the test set (Sewall, 1921). R^2 value ranges from -1 to $+1$. If $R^2 = -1$ or closer to -1 means that the ML model is not a good fit for the data set whereas $R^2 = +1$ or closer to $+1$ indicates the best performance of the model (Sewall, 1921). Hence, we try to tune the machine learning models to obtain a higher value of R^2.

3.3.2 Precision and recall

In classification problem, the models are usually trained with samples of same size from each category. However, it is not always the case. In this thesis, the dataset is manually labelled with two classes and the dataset becomes highly unbalanced. While dealing with unbalanced dataset, accuracy using confusion matrix (Figure 3.4) alone does not serve as an appropriate measure of the classifier's performance. For example, if 90 data points belong to class 1 and only 10 data points belong to class 2. If the classifier is able to classify all the class 1 and not class2 then the accuracy using confusion matrix will be 90% which is not correct. We want the classifier to classify class 2 data points as well. Performance measures like precision and recall are computed to correctly assess the performance of the classifier model. Figure 3.4 illustrates the confusion matrix. TP (true positive) and TN (true negative) are positive and negative tuples that were correctly classified respectively (Han, Kamber and Pei, 2012). FN (false negative) and FP (false positive) are misclassified as negative and positive respectively (Han, Kamber and Pei, 2012). Precision (Equation 3.4) is defined as the ability of the classifier to predict positive tuples correctly as positive considering all the tuples classified as positive (Han, Kamber and Pei, 2012). On the other hand, recall (Equation 3.5) is the ability of the classifier to predict positive tuples correctly as positive considering all the misclassified tuples (Han, Kamber and Pei, 2012).

$$Precision = \frac{TP}{TP + FP} \tag{3.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.5}$$

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	*P*
	no	*FP*	*TN*	*N*
	Total	P'	N'	$P + N$

Figure 3.4: Summary of confusion matrix (Han, Kamber and Pei, 2012).

3.3.3 Receiver operating characteristics (ROC) plot

The ROC curve is another method to evaluate the classifiers when the data set is unbalanced, similar to out case. The true positive rate (TPR) (Equation 3.6) is the fraction of positive tuples correctly classified by the model and false positive rate (FPR) (Equation 3.7) is the fraction of negative data points classified as positive (Han, Kamber and Pei, 2012). P and N are the number of positive and negative tuples. ROC curve is a plot of FPR and TPR where FPR is plotted on x-axis and TPR is plotted on y-axis. (Han, Kamber and Pei, 2012).

TPR and FPR of all the data tuples are plotted to form the ROC curve. The diagonal line in the Figure 3.5 shows denotes random guessing since it is difficult to infer that the

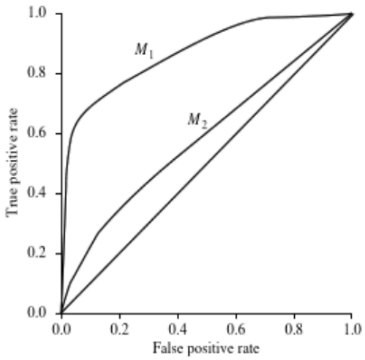


Figure 3.5: ROC curves of two classification models M_1 and M_2 (Han, Kamber and Pei, 2012).

model is leaning towards TPR or FPR (Han, Kamber and Pei, 2012). The more the model leans towards TPR the better classifier it is (Han, Kamber and Pei, 2012). If the model is closer to the diagonal, it means that the model performs poorly (Han, Kamber and Pei, 2012). Here, M_1 performs better than M_2 because initially M_1 has more TPRs as compared to M_2 (Han, Kamber and Pei, 2012). Also, M_2 is closer to the diagonal so it performs poorly (Han, Kamber and Pei, 2012).

$$TPR = \frac{TP}{P} \tag{3.6}$$

$$FPR = \frac{FP}{N} \tag{3.7}$$

Chapter 4

Methodology

As described in Chapter 2, diffusion is a physical phenomena modelled mathematically. It is a numeric value calculated using computer simulations. Furthermore, this numerical data as input is fed to machine learning algorithms described in Chapter **??** to produce diffusion prediction models. In order to answer the research questions specified in Chapter 1, it is required to make logical assessments based on strong mathematical reasoning. Following this approach, positivism research paradigm is chosen as the framework of the research design since it involves applying scientific methods to deliver the thesis objectives. Underpinning this paradigm, primary research is conducted to collect quantitative data from computer simulations (also known as computer experiments) and quantitative analysis is performed on it using machine learning algorithms. Firstly, data is collected using computer simulations and then fed into machine learning algorithms for further analysis.

4.1 Computer simulations of diffusion

Computer simulations are performed to generate two datasets. The simulation code is written in c and c++ programming languages. The simulations are conducted referring to Allen and Tildesley (1989).

4.1.1 Simulation of diffusion of nano-particles in a cubic array of repelling rods

Referring to the BD simulations conducted by Zhang, Hansing, Netz and Derouchey (2015), a cubic lattice (Figure 4.1) of repelling rods (attractive and repulsive forces) is used to represent the polymer chains. Diffusion of a single tracer particle is modelled in terms of the stochastic differential equation given by Cruz, Chinesta and Régnier (2012) as Equation 4.1.

$$dR(t) = \frac{1}{\xi}F(t)dt + BdW_t \tag{4.1}$$

Here, R(t) denotes the 3-dimentional position vector $R = (X, Y, Z)$ of the nano-particle, ξ is the friction coefficient, $B = \sqrt{2k_BT/\xi}$, $F = -\nabla U$ potential forces which represent the strength of the interaction of the diffusing particle with the polymer network, W_t is a three-dimensional Wiener process (Zhang, Hansing, Netz and Derouchey, 2015, Zhou and Chen, 2009). Wiener process is a continuous-time stochastic process and is a mathematical name given to the Brownian Motion (Allen and Tildesley, 1989). Referring to Zhang, Hansing, Netz and Derouchey (2015), Zhou and Chen (2009), the interaction between the diffusion particle

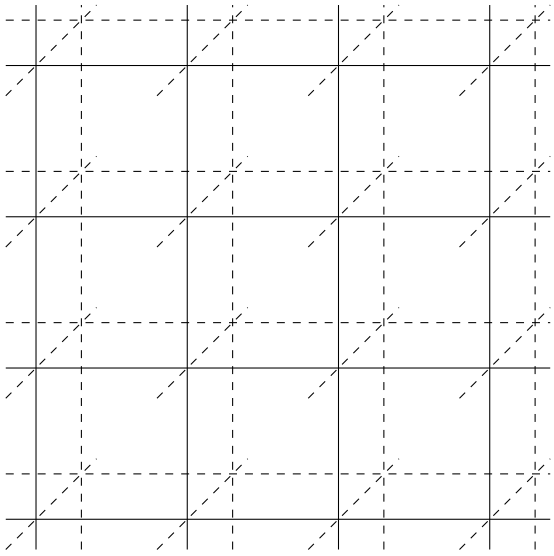


Figure 4.1: Illustration of the regular three-dimensional cubic network (infinitely extended in all three space dimensions).

and the polymer network can be modelled as a sum of steric and electrostatic contributions as Equation 4.2.

$$U = U^{\mathrm{s}} + U^{\mathrm{e}} \tag{4.2}$$

$$U^{\mathrm{s}}(r) = \begin{cases} 4\varepsilon \left[(\sigma/r)^{12} - (\sigma/r)^{6} + 1/4\right] & r \leq r_{\mathrm{c}} \\ 0 & r \geq r_{\mathrm{c}} \end{cases} \tag{4.3}$$

$$U^{\mathrm{e}}(r) = U_0 \exp\left[-r/k\right] \tag{4.4}$$

Here, s and e denote the steric and electrostatic interactions of the r is the distance between the centers of the particle and the network (Zhang, Hansing, Netz and Derouchey, 2015). With $r_{\mathrm{c}} = 2^{1/6}\sigma$ the steric interaction is purely repulsive (WCA potential) (Zhang, Hansing, Netz and Derouchey, 2015). The polymer grains are arranged in a regular cubic grid of mesh size a. The potential strength U_0 is the electrostatic interaction between the NP and the polymer lattice which can be positive (repulsive) or negative (attractive) (Zhang, Hansing, Netz and Derouchey, 2015). The Debye screening length is the range of the interaction of NP and the polymer network given by $k^2 = \epsilon\epsilon_0 k_B T/(2e^2 I)$ (Zhang, Hansing, Netz and Derouchey, 2015). Here, T denotes the temperature at which the diffusion occurs and k_B is the Boltzman constant and e is the elementary charge (Zhang, Hansing, Netz and Derouchey, 2015). $I = \frac{1}{2}\sum_j n_j z_j^2$ is the ionic strength where z is the valence of the salt ions, and n is their bulk number densities (Zhang, Hansing, Netz and Derouchey, 2015). It is assumed the tracer particle and network links are of the same diameter σ.

The quantities in the Equations 4.2, 4.3, 4.4 carry their physical units of time and length and so on. However, it is preferred to work with dimensionless (unitless) quantities in the simulations. The basic length scale is the diameter of the particle σ. Therefore, all lengths are measured in units of σ introducing dimensionless quantities $R^* = R/\sigma, r^* = r/\sigma$. The basic timescale is the diffusion time $\tau_{\mathrm{B}} = \sigma^2/D$ where $D = \sqrt{2T/\xi}$ is the single-particle diffusion coefficient. Thus, time is measured in units of τ_{B}, $t^* = t/\tau_{\mathrm{B}}$, $W^* = W/\sqrt{\tau_{\mathrm{B}}}$. Using thermal energy $k_B T$ as a basic energy scale, $\varepsilon^* = \varepsilon/(k_B T)$. The Dimensionless forces are therefore $F^* = \sigma F/(2k_B T)$. In terms of these dimensionless quantities, Equation 4.1 is written as

Equation 4.5.

$$dR^* = \frac{1}{\sigma} dR = \frac{\tau_{\mathrm{B}}}{\sigma\xi} \frac{k_B T}{\sigma} F^* dt^* + \sqrt{\frac{2 k_B T \tau_{\mathrm{B}}}{\sigma^2 \xi}}\, dW_t^* \\ = F^* dt^* + dW_t^* \tag{4.5}$$

The dimensionless forces $F^* = -\frac{\sigma}{2k_B T}\nabla U$ derived from the Equation 4.2 are given by Equations 4.6 and 4.7. Thus, there are four dimensionless model parameters as given in Equation 4.8 that determine the resulting properties, where $a^* = a/\sigma$ is the reduced mesh size, $\varepsilon^* = \varepsilon/(2T)$ the relative strength of Lennard-Jones (LJ) repulsion, $U_0^* = (U_0/2T)$ the relative strength of electrostatic interactions and $k^* = k/\sigma$ the dimensionless screening length.

$$F^{*,\mathrm{s}} = \varepsilon^* \frac{24}{r^*}[(r^*)^{-12} - 2(r^*)^{-6}]\hat{r} \tag{4.6}$$

$$F^{*,\mathrm{e}} = \frac{U_0^*}{k^*} e^{-r^*/k^*}\, \hat{r} \tag{4.7}$$

$$a^*, \varepsilon^*, U_0^*, k^* \tag{4.8}$$

Based on different combinations of values of these parameters, diffusion of tracer particles is computed as follows:

1. Initialize the parameters as per Table 4.1. Nt is the total number of time-step iterations of the simulations and the size of each time-step is Δt. $nparticles$ is the number of tracer particles in the cubic lattice whose diffusion is calculated. i_{out} is the step interval at which the output is logged in the output file.

Table 4.1: Parameter settings of BD simulation

Parameters	**Values**
Number of atoms (tracer particles) $(nparticles)$	100
Mesh size (a)	4, 6, 8, 10
Electrostatic Potential (U_0)	Range of [-20, +20]
Screening Length (k)	1, 2, 3, 4
Strength of LJ repulsion (ϵ)	0.5, 1, 2
Time-step (Δt)	0.1
Number of integration steps (Nt)	20
Skip number of step for output (i_{out})	10

2. **Compute new positions of NPs** : The Heun scheme is used to integrate the stochastic differential Equation 4.5. As per the Heun scheme, the next position of the particles are computed using the Euler-Maruyama given by Equation 4.9.

$$\bar{R}^* = R^*(t) + F^*(t)\Delta t + \sqrt{\Delta t}\,\zeta \tag{4.9}$$

where $F^*(t) = F^*(R(t))$ are the forces calculated with the current position and ζ a normally distributed random variable with zero mean and unit variance.

3. **Compute forces as per new positions** : All the forces (Brownian contribution, electrostatic potential, LJ force) acting on the particles in x, y and z directions in the current position are calculated. The Brownian contribution is given by $B = \sqrt{\Delta t}$. The polymer network is modelled as rigid rods, forming a perfect cubic grid. The particles interact with these rods. Consider the particle at position $= (X, Y, Z)$ with $|| < a$ and consider its interaction with the rod 1 given by $(s, 0, 0)$ and rod 2 given by $(s, 0, a)$ with $s \in R$. In principle, it is required to integrate the interaction potential along the rod. Instead, interaction potential is interpreted as Equation 4.2 giving the integrated potential when the distance vector is interpreted as the shortest distance. So in this case the distance vector to rod 1 is $r_1 = (0, Y, Z)$ and $r_2 = (0, Y, Z - a)$. Similarly, the interaction potential is calculated for all the rods. The LJ (steric) force is calculated using the Equation 4.3 and the electrostatic forces is calculated using the Equation 4.4. Calculate forces corresponding to these new positions as given in Step 2 by $\bar{F}^* = F^*(\bar{R}^*)$.

4. **Update positions** : Update the positions using Equation 4.10.

$$\bar{R}^*(t + \Delta t) = R^*(t) + \frac{1}{2}[F^*(t) + \bar{F}^*]\Delta t + \sqrt{\Delta t}\,\zeta \tag{4.10}$$

5. **Integrate forces** : An ensemble of N realisation of the stochastic process, meaning the integration of Equations 4.9, 4.10 is repeated with different random numbers ζ independently N times. Expectation values are then calculated by ensemble averages, $R^*(t) = N^{-1}\sum_{i=1}^{N} R_i^*(t)$. Solutions $R_W(t)$ to Equation 4.1 for given initial condition $R(0)$ and given realisations of the noise W_t are called 'trajectories'. For the same initial condition but different realisations different trajectories are obtained. Therefore, the expectation values are calculated as ensemble averages, $\langle A(R(t))\rangle = N^{-1}\sum_{i=1}^{N} R_{W^{(i)}}(t)$, with any function $A(R)$ of interest and $R_{W^{(i)}}$ the ith trajectory. The first quantity is the mean position, $A(R) = R$. Since the noise term is zero on average, an ordinary differential equation is given by Equation 4.11.

$$\frac{d}{dt}\langle R\rangle = \frac{1}{\xi}\langle F\rangle \tag{4.11}$$

If no external forces are present then $\langle F\rangle = 0$ and this quantity is at most a check of the numerical implementation. The primary quantity of interest is the f mean-square displacement, $A(R) = [R(t) - R(0)]^2$. From stochastic calculus we get, Equation 4.12 or, for each component as Equation 4.13.

$$\frac{d}{dt}\langle R(t)^2\rangle = \frac{2}{\xi}\langle R \cdot F\rangle + 3B^2 \tag{4.12}$$

$$\frac{d}{dt}\langle X(t)^2\rangle = \frac{2}{\xi}\langle X F_x\rangle + B^2 \tag{4.13}$$

In the absence of interactions, $F = 0$, free diffusion is given by Equation 4.14 with the single-particle diffusion coefficient $D = B^2 = 2T/\xi$.

$$\langle [X(t) - X(0)]^2\rangle = Dt \tag{4.14}$$

In the presence of interactions, $F \neq 0$, the mean-square displacement $\langle [R(t) - R(0)]^2\rangle$ is typically not a linear function of t and Equation 4.14 can not be applied naively. However, in many physical systems a **short-time diffusion coefficient** D_{short} is defined by fitting Equation 4.14 to the data for short enough times, $t < t_{\text{short}}$. Typically, $D_{\text{short}} = D$

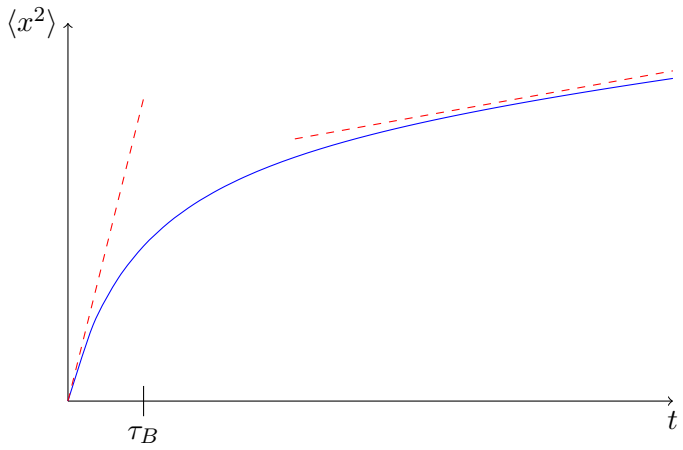


Figure 4.2: Schematic of mean-square displacement as function of time (blue). Red dashed lines indicate the short- and long-time diffusive regimes

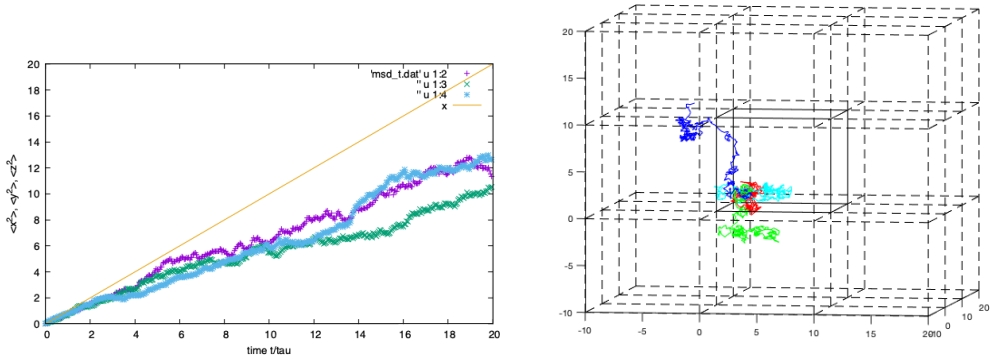


Figure 4.3: Left: The mean-square displacement $\langle x^2(t) \rangle, \langle y^2(t) \rangle, \langle z^2(t) \rangle$ as a function of t/τ_{B} for $a^* = U_0^* = 10, \varepsilon^* = 1, k^* = 2$ and ensemble of $N = 100$. The brown line denotes the non-interacting case. Right: Trajectories of four selected particles.

since, loosely speaking, the particle "did not have enough time to experience interaction forces". The **long-time diffusion coefficient** is defined by D_{long} by fitting Equation 4.14 to the data for long enough times, $t > t_{\mathrm{long}}$. While typically $t_{\mathrm{short}} \lesssim \tau_{\mathrm{B}}$, the value of t_{long} is not known in advance. Therefore, it is required to be careful and check whether the simulations are long enough to cover times larger than t_{long}. This is illustrated in Figure 4.2. Figure 4.3 shows the different components of the mean-square displacement $\langle R^* \cdot R^* \rangle$ and corresponding trajectories. Strong fluctuations are seen due to the small ensemble size ($N = 100$). Nevertheless, it is apparent that diffusion is slowed down (compared to free diffusion) due to the interaction with the network. If the network attracts the particles very strongly, diffusion almost comes to a halt. Therefore, defining and evaluating the diffusion coefficient becomes problematic as illustrated in Figure 4.4.

6. Repeat steps 2 to 5 for each Δt until Nt.

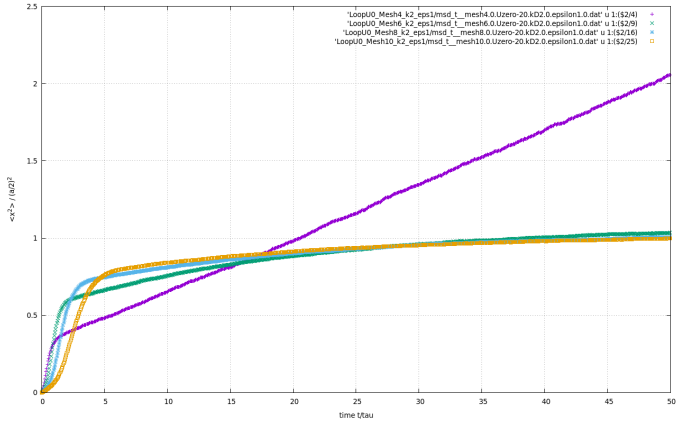


Figure 4.4: Mean-square displacement $\langle x^2 \rangle / (a/2)^2$ scaled with half the mesh size a squared for strongly attractive network, $U_0^* = -20$, $k^* = 2, \varepsilon^* = 1$.

4.1.1.1 Description of data set

The data set generated consists of 4 feature columns; mesh size (a), electrostatic potential (U_0), screening length (k), strength of LJ repulsion (ϵ) and 1 target column diffusion (D). All the values in the data set are of continuous data type. The total size of the data set is 924.

4.1.2 Molecular dynamics simulation of diffusion of magnetic NP in ferrofluids

Langevin dynamics simulation of ferrofuilds is conducted to study the effect of NP particle volume fraction (ϕ), dipolar coupling constant (λ) and the Langevin parameter (α) on diffusion of magnetic NP referring to Wang, Holm and Müller (2002). The translational and rotational Langevin equations of motion of MNP i are given by Equations 4.15 and 4.16 respectively. M_i and I_i are the mass and inertia tensor of the MNP, $\lceil_T$ and $\lceil_R$ are the translational and rotational friction constants respectively. ξ_i^T and ξ_i^R are Gaussian random force and torque respectively.

$$M_i \dot{v_i} = F_i - \lceil_T v_i + \xi_i^T \tag{4.15}$$

$$I_i \cdot \dot{\omega_i} = \tau_i - \lceil_R \omega_i + \xi_i^R \tag{4.16}$$

Substituting, Equations 4.19 and 4.20 into Equations 4.15 and 4.16, we obtain the dimensionless equations of motions given by Equations 4.17 and 4.18.

$$\dot{v_i^*} = \sum_{j \neq i} (F_{ij}^{dip*} + F_{ij}^{LJ*}) - \lceil_T^* v_i^* + \xi_i^{T*}) \tag{4.17}$$

$$I_i^* \cdot \dot{\omega_i^*} = \sum_{j \neq i} \tau_{ij}^{dip*} + m_i^* \times H^* - \lceil_R^* \omega_i^* + \xi_i^{R*} \tag{4.18}$$

Here, the dimensionless quantities are given by length $r^* = r/\sigma$, dipole moment $m^{*2} = m^2/4\pi\mu_0 \in \sigma_3$, moment of inertia $I^* = I/(M\sigma^2)$, time $t^* = t(\in /M\sigma^2)^{1/2}$, the friction constants $\lceil_T^* = \lceil_T(\sigma_2/M \in)^{1/2}$ and $\lceil_R^* = \lceil_R/(M\sigma_2 \in)^{1/2}$, magnetic field $H^* = H(4\pi\mu_0\sigma_3/ \in)^{1/2}$, temperature $T^* = kT/ \in$.

1. Initialize model parameters as per Table 4.2.

Table 4.2: Parameter settings of MD simulation

Parameters	**Values**
Number of atoms (tracer particles) N	1000
Moment of inertia I^*	0.4
Friction constant $\lceil_T^*$	10
Friction constant $\lceil_R^*$	3
temperature T^*	1
Time-step (Δt^*)	0.002
Number of integration steps (Nt)	200,000

2. **Compute forces** : The ferrofluid model consists of N spherical MNPs of diameter σ. They are distributed in a cubic box of side length L. Each MNP possesses a dipole moment m_i at its center. Adopting periodic boundary conditions, the dipole-dipole interaction potential between particle i and j is given by Equation 4.19.

$$U_{ij}^{dip} = \frac{1}{4\pi\mu_0} \sum_{n \in Z^3} \left\{ \frac{m_i \cdot m_j}{|r_{ij} + nL|^3} - \frac{3[\, m_i \cdot (\, r_{ij} + nL)\,]\,[\, m_j \cdot (\, r_{ij} + nL)\,]}{|r_{ij} + nL|^5} \right\} \tag{4.19}$$

Here, $r_{ij} = r_i - rj$ is the displacement vector of the two particles. The sum is performed over all cubic lattice points, $n = (n_x, n_y, n_z)$ with n_x, n_y, n_z integers. The LJ potential is adopted to model the short-range steric interaction potential between the MNPs given by Equation 4.20.

$$U_{ij}^{LJ} = 4\varepsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} - C(R_c) \right] \tag{4.20}$$

Here, $C(R_c) = (\sigma/R_c)^{12} - (\sigma/R_c)^6$ with a cutoff radius of $R_c = 2^{1/6}\sigma$.

3. **Integrate equations of motions** : Using Ewald summation for the long-range dipole-dipole interactions, Equation 4.19 is evaluated by Equation 4.21.

$$U_{ij}^{dip} = U_{ij}^{(r)} + U_{ij}^{(k)} + U_{ij}^{(self)} + U_{ij}^{(surf)} \tag{4.21}$$

Here, the $U_{ij}^{(r)}$ is the real-space, $U_{ij}^{(k)}$ is k space, $U_{ij}^{(self)}$ is self and $U_{ij}^{(surf)}$ is the surface. These contributions are given by Equations 4.22, 4.23, 4.24 and 4.25 where $B(r)$ and $C(r)$ are given by Equations 4.26 and 4.27 respectively.

$$U_{ij}^{(r)} = \frac{1}{4\pi\mu_0} \sum_{n \in Z^3} \{(\, m_i \cdot m_j)\, B(\, |r_{ij} + n|) \, - [\, m_i \cdot (\, r_{ij} + n)\,] \times [\, m_i \cdot (\, r_{ij} + n)\,]\, C(\, |r_{ij} + n|)\,\} \tag{4.22}$$

$$U_{ij}^{(k)} = \frac{1}{4\pi\mu_0 L_3} \sum_{k \in Z^3, k \neq 0} \frac{4\pi}{k^2} exp[\, -(\, \pi k/\kappa L)^{\,2}]\,(\, m_i \cdot k) \, \times (\, m_j \cdot k)\, exp(\, 2\pi i k \cdot r_{ij}/L) \tag{4.23}$$

$$U_{ij}^{(self)} = -\frac{1}{4\pi\mu_0} \frac{2\kappa^3}{3\sqrt{\pi}} (\, m_i^2 + m_j^2) \tag{4.24}$$

$$U_{ij}^{(surf)} = \frac{1}{4\pi\mu_0} \frac{4\pi}{(\, 2\mu_{BC} + 1)\, L^3} m_i \cdot m_j \tag{4.25}$$

$$B(r) = [\, erfc(\kappa r) + (2\kappa r/\sqrt{\pi})\, exp(-\kappa^2 r^2)\,]/r^3 \tag{4.26}$$

$$C(r) = [\, 3erfc(\kappa r) + (2\kappa r/\sqrt{\pi})\,(3 + 2\kappa^2 r^2) \times exp(-\kappa^2 r^5)\,]/r^3 \tag{4.27}$$

The inverse length κ is the splitting parameter of the Ewald summation. Leap-frog method is used to solve the equations of motions (Equations 4.17 and 4.18).

4. Repeat steps 2 and 3 for each Δt^* until Nt.

4.1.2.1 Description of data set

The data set generated consists of 3 feature columns; particle volume fraction (ϕ), dipolar coupling constant (λ), langevin parameter (α) and 3 target columns; diffusion (D), average cluster size (savg1), magnetization (M). However, in this thesis, only diffusion (D) is considered as the target. All the values in the data set are of continuous data type. The total size of the data set is 90.

4.2 Implementation of machine learning algorithms

The data set is fed to machine learning algorithms mentioned in Chapter 3. It is implemented in python language. It is a very popular language used for data science. It is powerful due to the extensive libraries such as numpy, pandas, seaborn, matplotlib to read, manipulate and perform analysis on the data (McKinney, 2013). Dedicated machine learning python libraries scikit-learn (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot and Duchesnay, 2011) and scipy (Virtanen, Gommers, Oliphant, Haberland, Reddy, Cournapeau, Burovski, Peterson, Weckesser, Bright, van der Walt, Brett, Wilson, Millman, Mayorov, Nelson, Jones, Kern, Larson, Carey, Polat, Feng, Moore, VanderPlas, Laxalde, Perktold, Cimrman, Henriksen, Quintero, Harris, Archibald, Ribeiro, Pedregosa, van Mulbregt and SciPy 1.0 Contributors, 2020) make it extremely convenient to train and test different models. Google colaboratory (Bisong, 2019) provides an integrated development environment to write and execute code in python. It provides a flexibility to sync the .ipynb code file with the google drive. It provides a convenient way to segment code, add rich text markdowns which can incorporate latex, html, image and so on. The .ipynb file containing entire code is uploaded on the gitlab repository (Desai, 2021).

Since it is not possible to determine which ML algorithms will learn better the given data set, different descriptive (EDA, pearson coefficient, feature analysis, PCA, t-Test) and predictive (random forest, KNN, decision tree, extreme gradient bossting and k-means) ML algorithms are used and their results are compared. To evaluate the models in case of regression, 10-fold cross validation is performed 30 times which is also known as repeated k-fold cross validation. It is the best technique to assess the performance of ML models (Nakatsu, 2021). In 10-fold cross validation, the data set is spilt into 10 partitions 10 times. Each time 1 (10%) part is selected as test set and the remaining 9 parts (90%) are used to train the model. This ensures that every data point is part of both the training and test set. The average R^2 score over the 10 folds is calculated and saved. This process is again repeated 30 times to make sure that the high R^2 score is not generated just because a random split of the data set is good. The average R^2 score is over 30 iterations is computed and considered as the final score of the model. Therefore, each model is evaluated for a total of $30 \times 10 = 300$ times before arriving at the final score. In case of classification, hold-out method (Tan, Steinbach and Kumar, 2016) is performed where the models are trained on the 70% of the data and

tested on the remaining 30%. This is repeated 30 times and average the precision and recall scores are computed.

Feature analysis is performed by using flexible neural tree (FNT) software provided by Ojha (2016). The FNT model is trained and tested on the diffusion data set using 10 cross validation with the relevant settings mentioned in Chapter 3. This experiment is manually repeated 30 times. The selection of each feature is recorded for every iteration. If a feature is selected then it is recorded as 1 otherwise 0. Similarly, this is performed for all the combinations of 2 and 3 selected features together in every iteration. The details of this experiment is maintained in an excel file uploaded on the gitlab (Desai, 2021). The average R^2 score and the selection rate is given by the Equation 4.28 is calculated over 30 iterations. Further, FNT analysis file is read in python to perform t-test on the results.

$$\text{Selection Rate} = \frac{\text{Number of times the individual feature or feature subset is selected} \times 100}{30} \tag{4.28}$$

Chapter 5

Results and discussion : Diffusion of NP in polymer

5.1 Descriptive analysis

5.1.1 Exploratory data analysis

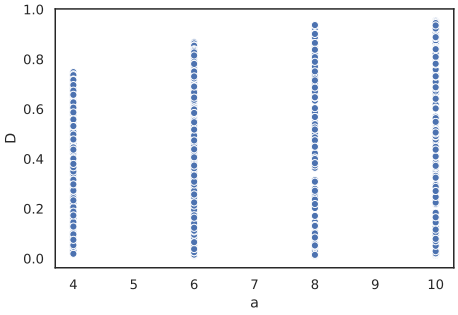
As explained in Chapter 3, electrostatic potential (U_0), mesh size (a), screening Length (k) and strength of LJ repulsion (ϵ) influence the rate of diffusion (D). In order to understand the nature and strength of the influence, the first step is to visualize on a simple scatter-plot as shown in Figure 5.1. No clear pattern or relationship between a, k, ϵ and D can be established using a simple scatter-plot. However, some pattern seems to exist in the influence of U_0 on D. Especially at $U_0 = 0$, there is free diffusion in the absence of electrostatic potential. During free diffusion, nano-particles diffuse freely in the absence of any external force and hence the diffusion is highest at $U_0 = 0$.

5.1.2 Pearson correlation coefficient

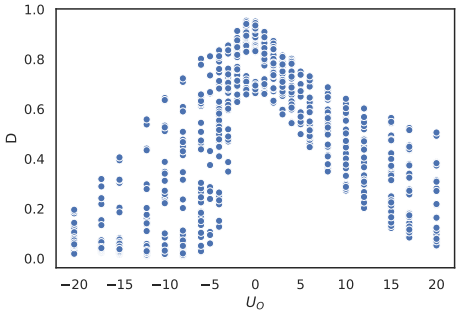
During the simulation, each feature value is manually entered to produce a value of D. Hence, finding a correlation between the feature values does not hold any meaning. The correlation coefficient is calculated between each feature and D. Referring to Table 5.1, U_0 has a positive correlation with D, diffusion increases with U_0. Usually, diffusion is found to increase with larger a. On the contrary, a has a negative correlation with D. As mesh size increases the diffusion decreases which is an unexpected result. However, this could be due to the influence of U_0 on diffusion.

Table 5.1: Pearson correlation coefficient of all the features with respect to D

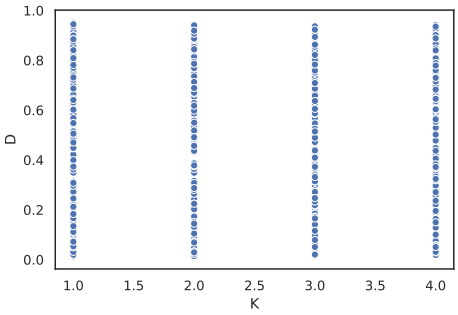
Feature	**Pearson Coefficient**
Mesh Size (a)	-0.07
Electrostatic Potential (U_0)	0.27
Screening Length (k)	0.08
Strength of LJ repulsion (ϵ)	0.01



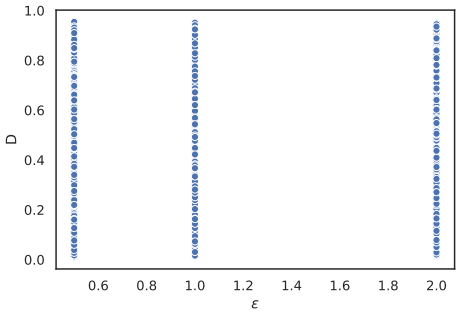
(a) a vs. D



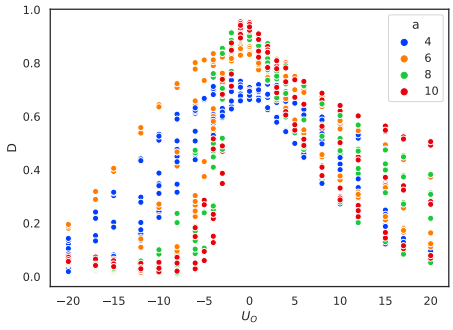
(b) U_0 vs. D



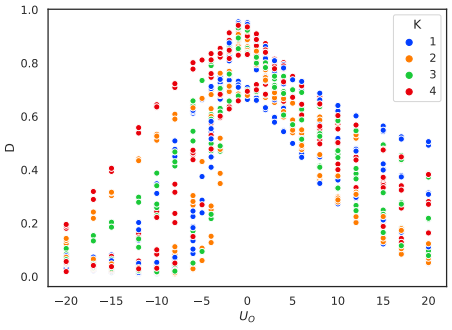
(c) k vs. D



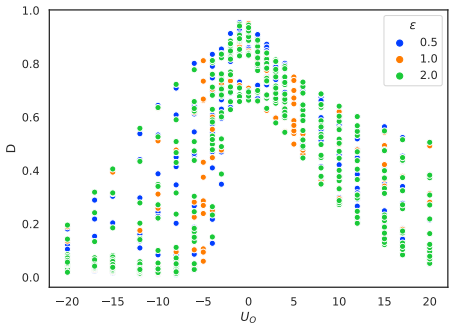
(d) ϵ vs. D



(e) a, U_0 vs. D



(f) k, U_0 vs. D



(g) ϵ, U_0 vs. D

Figure 5.1: Variation of diffusion (D) with respect to electrostatic potential (U_0), mesh size (a), screening length (K) and strength of LJ repulsion (ϵ)

To further investigate this behaviour, a, U_0, D are visualized together using a scatter-plot in Figures 5.1(e), (f), (g). It can be inferred that the diffusion of particles increases even for lower mesh sizes ($a = 4, a = 5$) when U_0 is negative (attractive forces). In case of free diffusion, it is high for large mesh sizes ($a = 6, a = 8, a = 10$) and low for small mesh size ($a = 4$). To summarize, the rate of diffusion is not only affected by each individual features but also by a combination of them.

5.1.3 Principal component analysis

Four dimensional data consisting of U_0, a, k, ϵ are reduced to two dimensional principal components 0 and 1 using PCA. Principal components 0 (PC0) and 1 (PC1) account for 95% and 4% of the total data variation respectively. Referring to the Table 5.2, PC0 can be used to measure U_0 since the value of loading is -1. Consequently, higher value of PC0 implies low value of U_0. Referring to Figure 5.2, it is evident that for high U_0, the diffusion is low. PC1 can be used to mainly measure a due to high positive loading. Therefore, high PC1 values imply large a and presence of some k and ϵ. As per Figure 5.2, diffusion is high for both high and low PC1 values. Hence, the relationship between a and D is not clear.

Table 5.2: Influence of U_0, a, k, ϵ on the principal components 0 and 1

Feature	Principal Component 0	Principal Component 1
a	≈ 0	0.99
U_0	-1	≈ 0
k	≈ 0	0.03
ϵ	≈ 0	0.0008

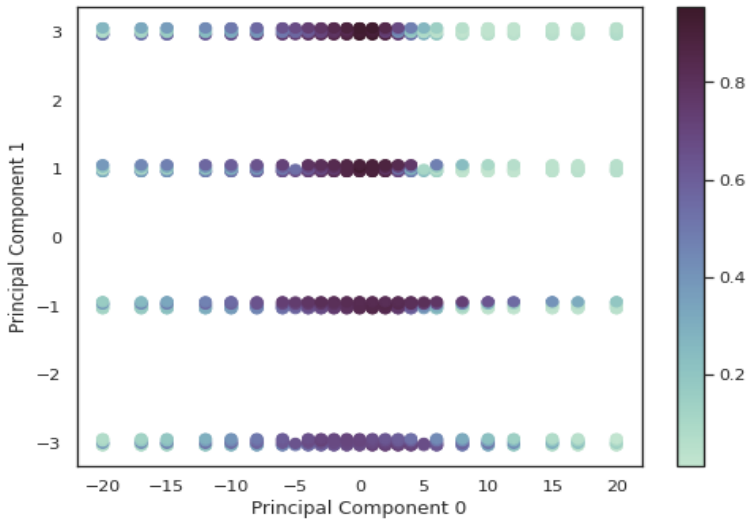


Figure 5.2: Variation of diffusion with respect to principal components

5.1.4 Feature Selection

Table 5.3: Feature ranking based on selection rate using flexible neural tree

Feature	**Selection Rate (%)**	**Rank**
Electrostatic Potential (U_0)	100	1
Mesh Size (a)	83.33	2
Screening Length (k)	60	3
Strength of LJ repulsion (ϵ)	30	4

Table 5.4: 2-feature subset selection using flexible neural trees

Features	**Selection Rate (%)**
a, U_0	83.33
U_0, k	60
a, k	53.33
U_0, ϵ	30
a, ϵ	23.33
k, ϵ	23.33

Table 5.5: 3-feature subset selection using Flexible Neural Trees

Features	**Selection Rate (%)**
a, U_0, k	53.33
U_0, k, ϵ	23.33
a, U_0, ϵ	23.33
a, k, ϵ	20

Table 5.3 shows the rank of the individual features based on the selection rate. Range of the rank is 1 to 5, 1 being the highest selection rate and 5 being the lowest selection rate. U_0 has the highest ranking which implies that U_0 is naturally selected by FNT in every iteration to predict diffusion and is the most important feature in the feature space. ϵ is the least important feature as it is selected only 30% of the time. Referring to Table 5.4, a and U_0 are selected together most of the time as compared to other combination of the features. On the other hand, a, U_0, k are selected more number of times as compared to other feature combinations. These results are consistent with those mentioned in the previous sections.

5.1.5 t-test

A random experiment is performed in Section 5.1.4 using FNT. In order to confirm that the results are not a mere coincidence, t-test is performed on different combination of feature selection data as illustrated in the Table 5.6. In case of a and U_0 t value is small and the p-value is greater than 0.05. Therefore, the null hypothesis is accepted. There is statistically significant evidence that a and U_0 belong to the same distribution. It implies that FNT naturally choosing a and U_0 together 83.33% of the times is not by chance and the results are actually correct. For the remaining cases, the t-value is high and p-value less than 0.05. This

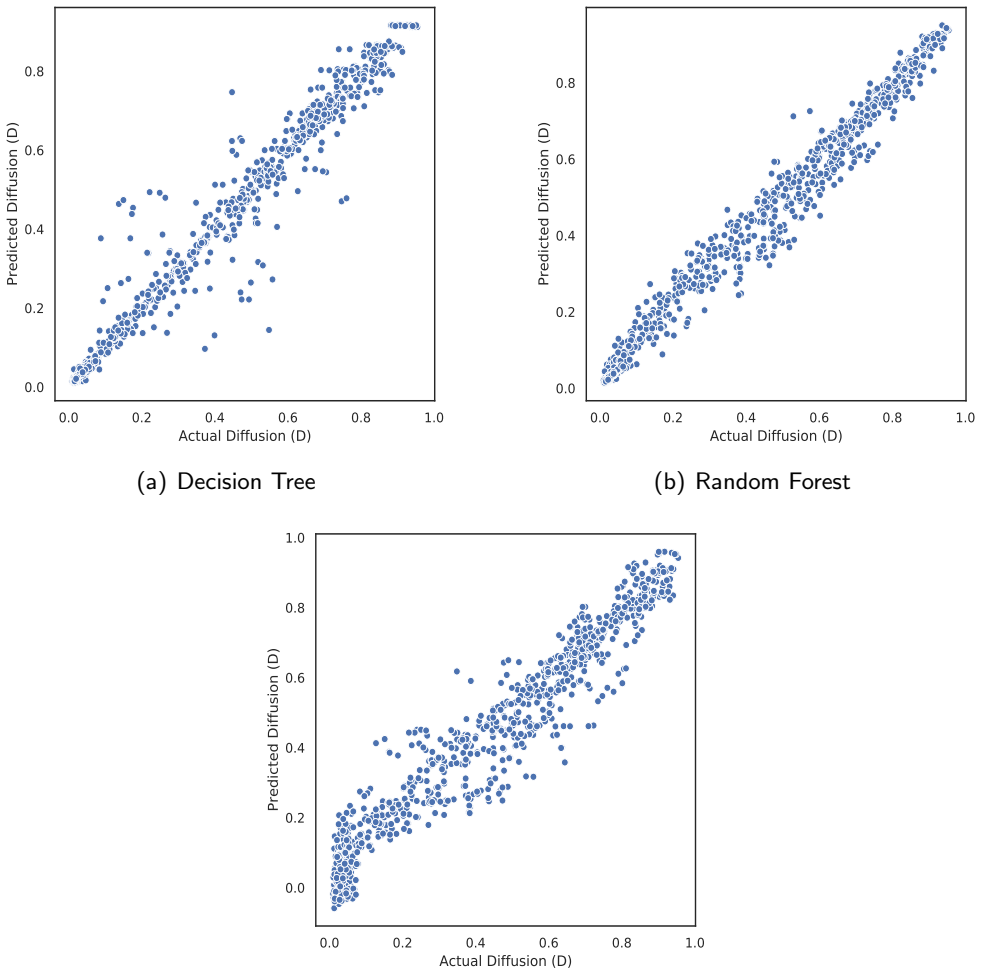
means that the features in each pair do not belong to the same distribution, thus, rejecting null hypothesis.

Table 5.6: t-test on FNT experiment data

Feature 1	Feature 2	T value	p-value
a	U_0	-0.84	0.40
a	ϵ	5.25	2.32×10^{-6}
U_0	k	4.14	10^{-4}
U_0	ϵ	7.94	7.71×10^{-11}

5.2 Predictive analysis

5.2.1 Regression



(a) Decision Tree

(b) Random Forest

(c) XGBoost

Figure 5.3: Actual vs predicted diffusion

The results from feature analysis provide the best features subsets. Using these results, decision tree, random forest and extreme gradient boosting algorithms are trained with 2 features, 3 features and all the 4 features. 30 times 10-fold cross validation is performed and the average R^2 over 30 iterations is computed. R^2 scores of all the models are compared as shown in Table 5.7. Random forest seems to learn the data set very well since it has the highest score. Figures 5.3(b) and 5.5 show that there are very few outliers (denoted by the red peaks) and the predicted diffusion values are almost the same as the actual values. In case of 4 features, the second highest score is that of decision tree. Figures 5.3(a) and 5.4 show that there are slightly more outliers (denoted by the red peaks) as compared to random forest. However in case of 3 features the accuracy of DT becomes equivalent to the random forest which remains unchanged. Furthermore, Figure 5.7 shows that the variance of the decision tree significantly reduces without ϵ. This could be due to the fact that ϵ is misleading the models to provide skewed predictions. Figures 5.3(c) and 5.4 show that the number of outliers is significantly more as compared to random forest. In this case as well, R^2 score is not affected by number of features. Interestingly, in case of 2 features, the accuracy scores of all the models decrease significantly.

Table 5.7: Comparison of R^2 scores of n feature subsets where $n = 2, 3, 4$

Algorithm	a, U_0	a, U_0, k	a, U_0, k, ϵ
Decision Tree	0.84	0.98	0.97
Random Forest	0.85	0.98	0.98
Extreme Gradient Boosting	0.86	0.94	0.94

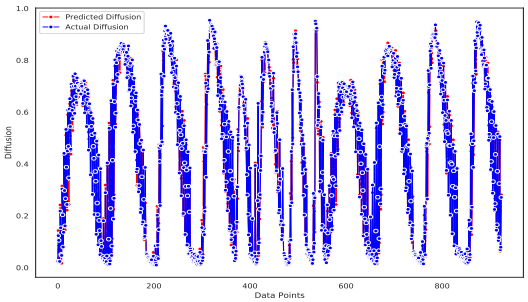


Figure 5.4: Decision tree

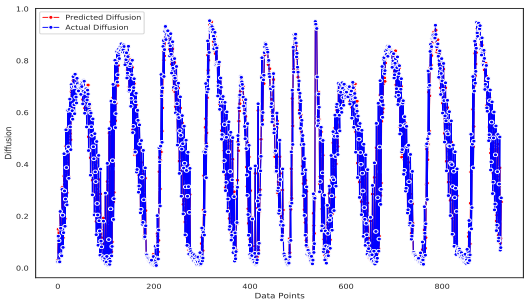


Figure 5.5: Random forest

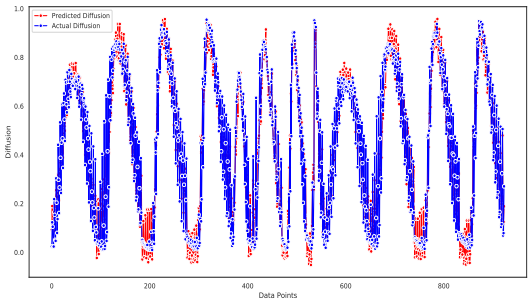


Figure 5.6: Extreme gradient boosting

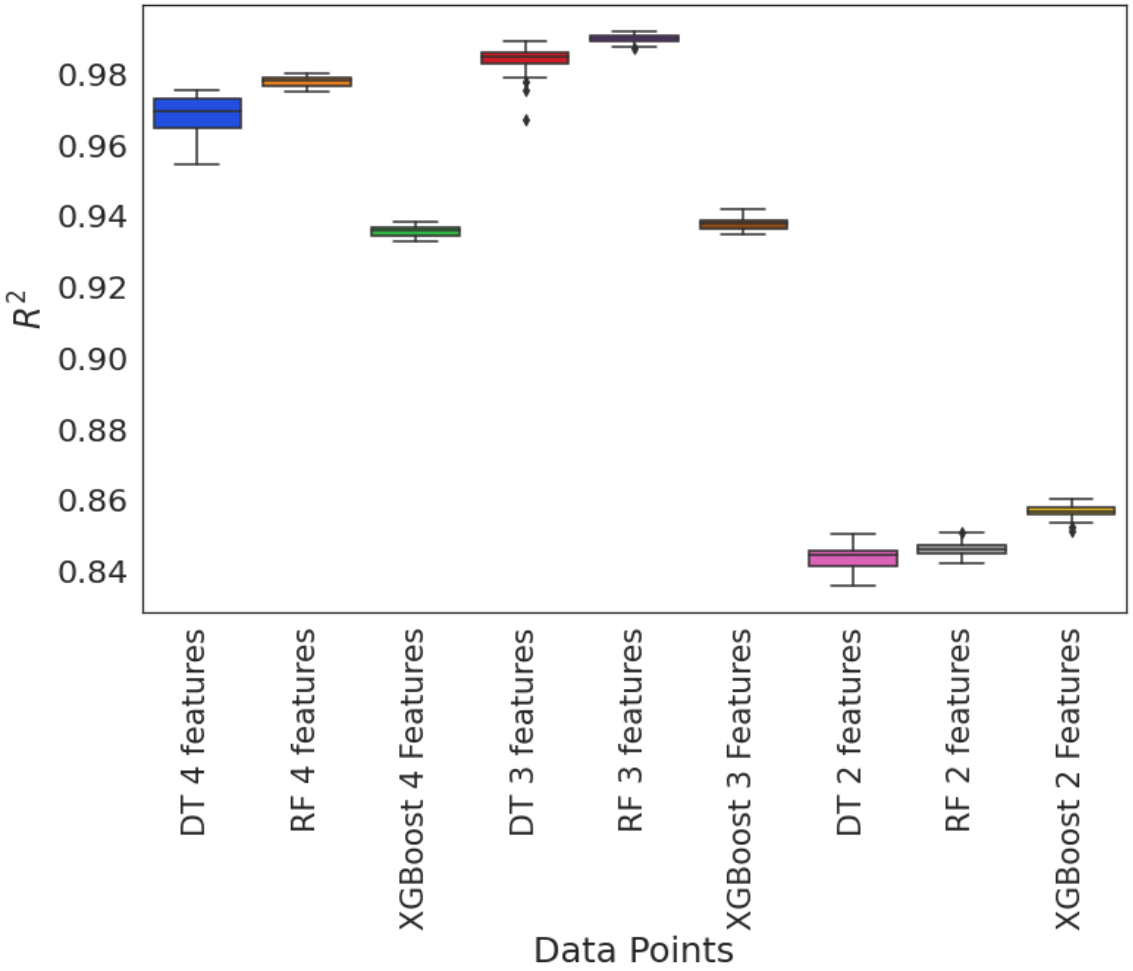


Figure 5.7: Comparison of R2 scores of Random Forest (RF), Decision Tree (DT) and XGBoost algorithms based on number of features

5.2.2 K-means and manual clustering

In order to apply k-means clustering to the data set, it is required to choose an appropriate k. This thesis uses Silhouette score to select the appropriate k. Silhouette score is computed for all the cluster sizes ranging from 1-10 as shown in Figure 5.8. It is evident that the Silhouette score is the least in case of $k = 9$. Following this, k-means algorithm is applied to the data set and it is divided into 9 clusters. However, it is observed that the clusters are formed based on

U_0 and not diffusion as shown in Figure 5.9. Moreover, the k seems to be very large and the clusters overlap even in case of U_0. This is not an expected outcome because the intention is to cluster the data set based on diffusion to extract some pattern associated with it.

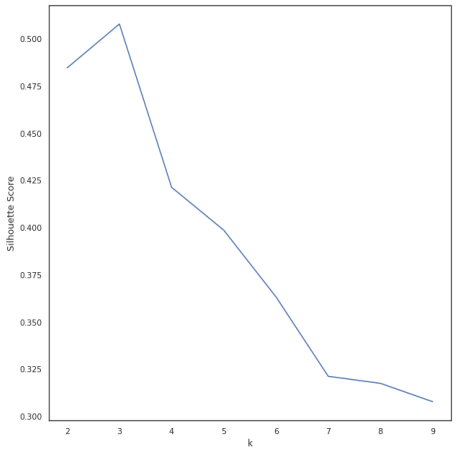
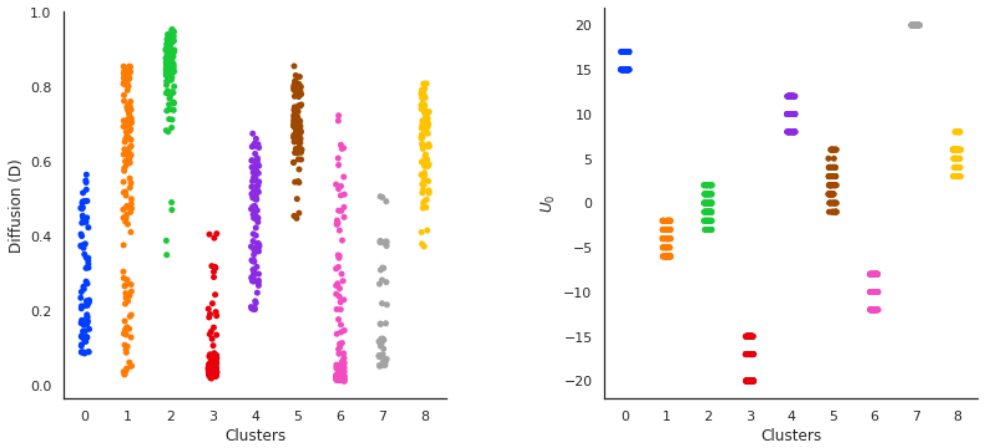


Figure 5.8: Silhouette score for k clusters



(a) Clusters vs. Diffusion (D)

(b) Clusters vs. U_0

Figure 5.9: Behaviour of k-means with respect to U_0 and D

To achieve the desired clusters, the data set is manually split into 2 clusters based on the histogram of diffusion as shown in Figure 5.10. It shows that many data points fall below $D = 0.08$ and using this information, the data is divided into 2 clusters; Class-1 corresponds to low diffusion ($D <= 0.08$) and Class-2 corresponds to high diffusion ($D > 0.08$). The manual clusters are plotted on the PCA grid in Figure 5.11 shows some overlap of both the clusters. This means that the even manually created clusters cannot be clearly separated.

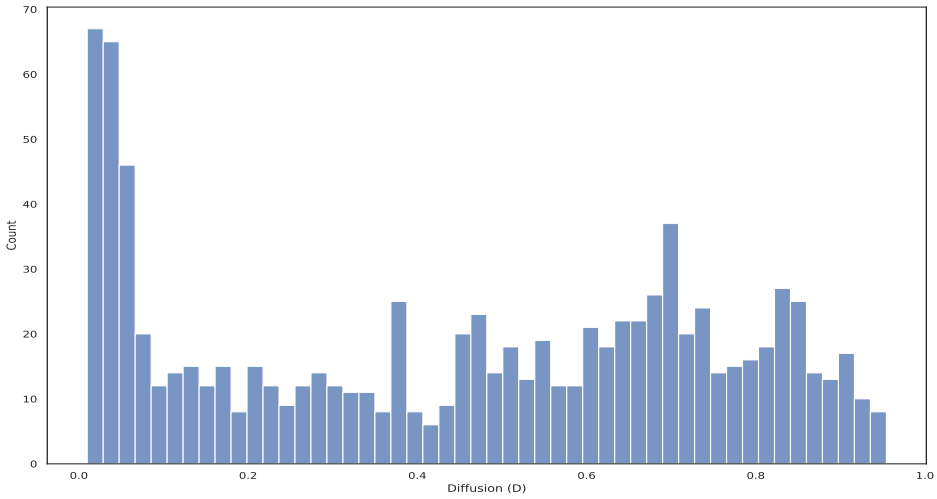


Figure 5.10: Histogram of diffusion

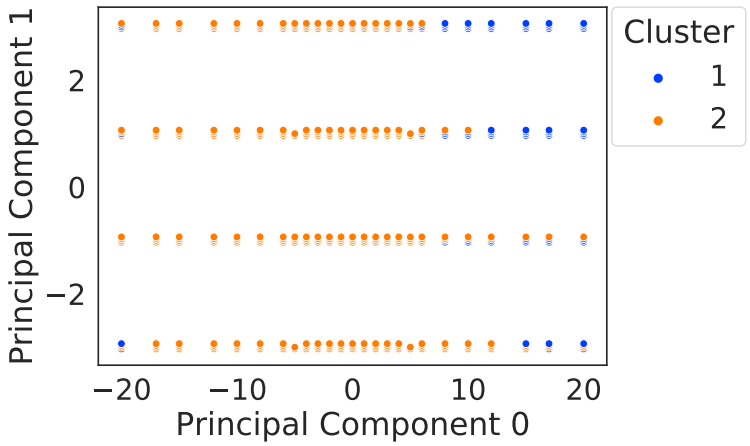


Figure 5.11: Manual clusters plotted on PCA grid

5.2.3 Classification

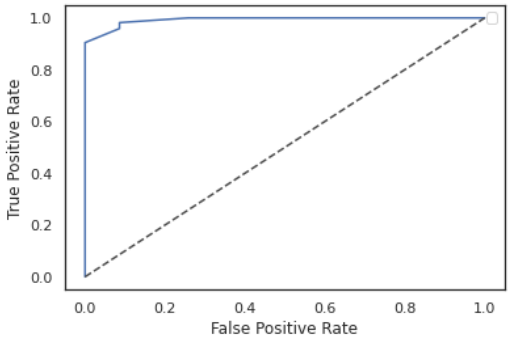
The original data set now also includes the discrete cluster labels generated in Section 5.2.2. Random forest (RF) and k-nearest neighbour (KNN) classification models are applied to the clustered data set. The size of Class-1 and Class-2 is 192 and 732 respectively. Since the data set is highly unbalanced, performance measures such as precision, recall and ROC are used to evaluate the models. RF and KNN models are trained using 2 types of training. In Type-1 training, the models are trained on 70% of cluster-1 and 70% of cluster-2 and tested on the remaining data. In Type-2 training, the models are trained on 70% of cluster-1 and tested on the remaining data. This process is repeated 30 times and the average precision and recall scores are computed as shown in Tables 5.8 and 5.9.

Table 5.8: Precision (P) and recall (R) scores of random forest

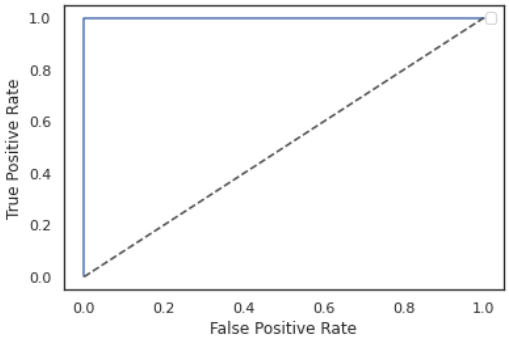
Features	Type of training	$D <= 0.08$		$D > 0.08$	
		P	**R**	**P**	**R**
a, U_0, ϵ, k	Type-1	99.83	99.82	99.95	99.95
	Type-2	7.34	100	0	0
a, U_0, k	Type-1	99.89	99.94	99.89	99.97
	Type-2	7.34	100	0	0

Table 5.9: Precision (P) and recall (R) scores of k-nearest neighbour

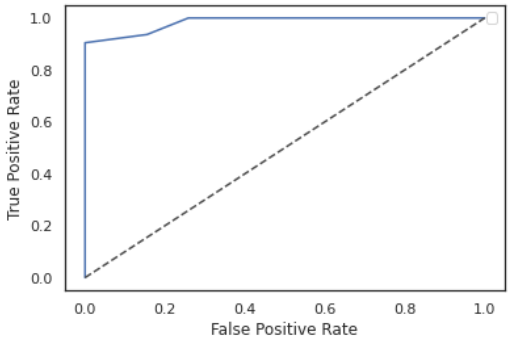
Features	Type of training	$D <= 0.08$		$D > 0.08$	
		P	**R**	**P**	**R**
a, U_0, ϵ, k	Type-1	85.58	87.3	96.65	96.03
	Type-2	7.34	100	0	0
a, U_0, k	Type-1	87.86	87.64	96.75	96.67
	Type-2	7.34	100	0	0



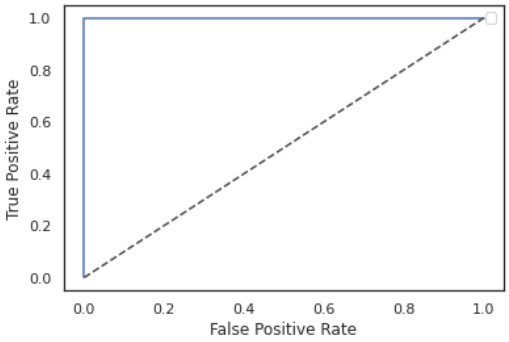
(a) KNN Type-2 training, 3 features



(b) RF Type-2 training, 3 features



(c) KNN Type-2 training, 4 features



(d) RF Type-2 training, 4 features

Figure 5.12: ROC curve of KNN and RF

It is observed that RF performs better than KNN in case of Type-1 training due to high precision and recall scores. This could be due to the ensemble method used by RF. The base

learners in RF may predict the class label incorrectly but RF's output is the majority class voted by all the base learners rather than a single learner. Since it is rare for majority base learners to predict incorrect class label, RF is able to overcome bias. In case of Type-2 training, the results for both the models are same. This could be due to the fact that they are trained with data points containing only Class-1. Therefore, they are able to classify Class-1 correctly but are not able to recognise Class-2. Better results are obtained for 3 features as compared to 4 features. These results supplement those in Section 5.2 that removal of ϵ improves the performance of ML models. In case of Type-2, the results remain same for RF and KNN. Figure 5.12 shows that in case of Type-2 training, 3 and 4 features, the ROC curve of KNN is closer to the diagonal line than that of RF. This means that RF performs better than KNN.

Chapter 6

Results and Discussion : Diffusion in ferrofluids

6.1 Descriptive analysis

6.1.1 Exploratory data analysis

As explained in Chapter 3, particle volume fraction (ϕ), dipolar coupling constant (λ) and Langevin parameter (α) influence the rate of diffusion (D). In order to understand the nature and strength of the influence, the first step is to visualize on a simple scatter-plot as shown in Figure 6.1. No clear pattern or relationship between α, ϕ and D can be established using a simple scatter-plot. However, there seems to be some pattern in influence of λ on D. For lower values of λ ($\lambda = 1, \lambda = 2$), D seems to be high whereas for the higher value of λ ($\lambda = 4$), the diffusion seems to be low.

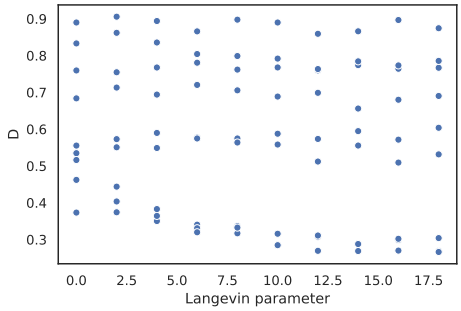
6.1.2 Pearson correlation coefficient

Correlation mathematically quantifies the magnitude and direction of the relationship between two variables. During the simulation, each feature value is manually entered to produce a value of the diffusion coefficient (D). Hence, finding a correlation between the feature values does not hold any meaning. The correlation coefficient is calculated for each feature and the diffusion coefficient (D). Referring to Table 6.1, all the 3 features have a negative correlation with D. However, the λ has the highest negative correlation with D as seen previously. As dipolar coupling constant increases diffusion decreases.

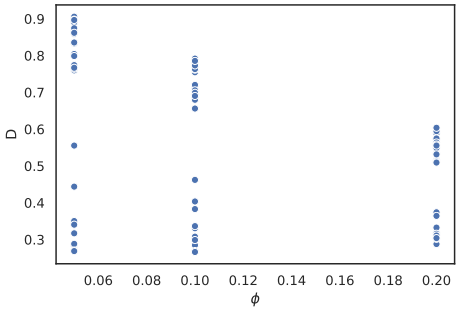
Table 6.1: Pearson correlation coefficient of all the features with respect to D

Feature	**Diffusion (D)**
Particle volume fraction (ϕ)	-0.38
Dipolar coupling constant (λ)	-0.85
Langevin parameter (α)	-0.1

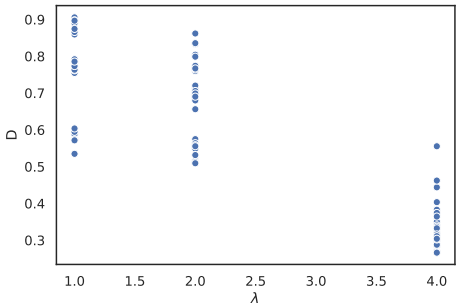
To further investigate this behaviour, α, ϕ, λ and D are visualized together using a scatter-plot as shown in Figure 6.1(c). It can be inferred that the range of diffusion slightly decreases for lower values of ϕ.



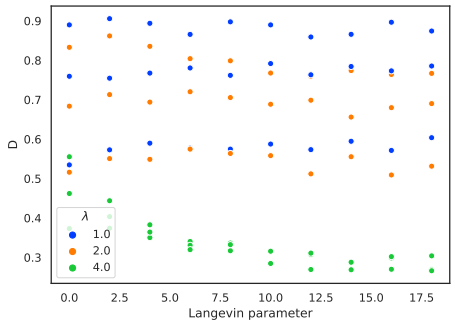
(a) α vs D



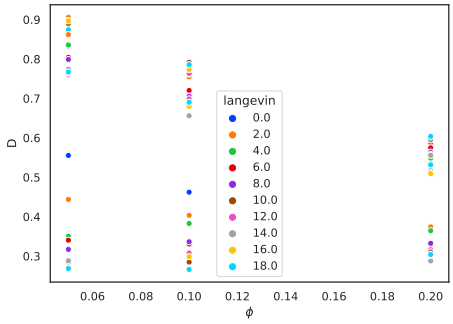
(b) ϕ vs D



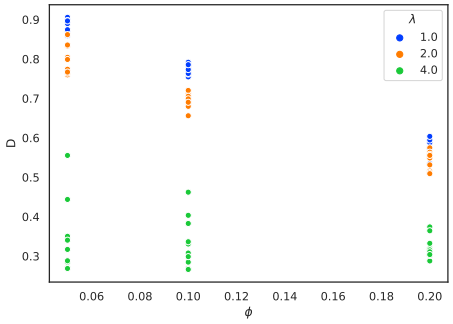
(c) λ vs D



(d) α, λ vs. D



(e) α, ϕ vs. D



(f) λ, ϕ vs. D

Figure 6.1: Variation of particle volume fraction (ϕ), dipolar coupling constant (λ) and Langevin parameter (α) with respect to diffusion (D)

6.1.3 Principal component analysis

Three dimensional data consisting of α, ϕ, λ are reduced to two dimensional principal components 0 and 1. Principal components 0 (PC0) and 1 (PC1) account for 95% and 5% of the total data variation respectively. Referring to the Table 6.2, PC0 can be used to measure α since the value of loading is positive 1. Consequently, higher value of PC0 implies high value of α. Referring to Figure 6.2, the relationship between α and D is not clear. PC1 can be used to mainly measure λ due to high positive loading. Therefore, high PC1 values imply large λ. As per Figure 6.2, diffusion is low for high PC1 values which is contradicting. Hence, the relationship between λ and D is not clear. Lastly, ϕ does not seem to be an important feature

according to PCA.

Table 6.2: Influence of ϕ, λ, α on the principal components 0 and 1

Feature	Principal Component 0	Principal Component 1
ϕ	0	0
λ	≈ 0	1
α	1	≈ 0

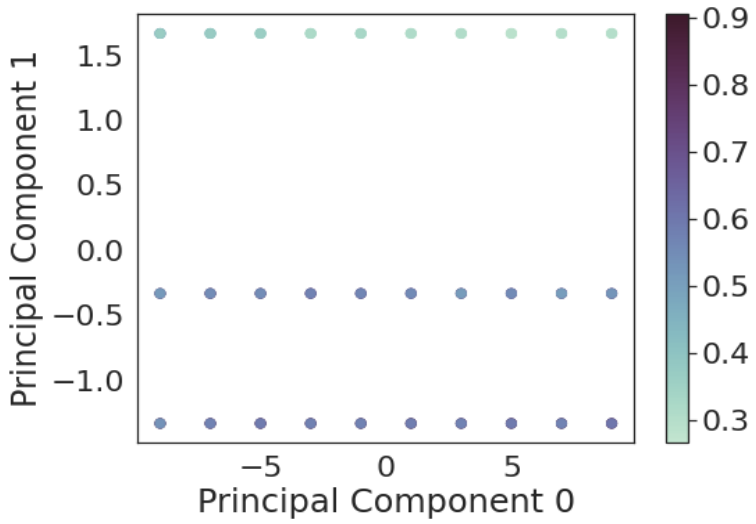


Figure 6.2: Variation of diffusion with respect to principal components

6.1.4 Feature selection

Table 6.3 shows the rank of the individual features based on the selection rate. Range of the rank is 1 to 5, 1 being the highest selection rate and 5 being the lowest selection rate. ϕ, λ have the highest ranking which implies that ϕ, λ are naturally selected by FNT in every iteration to predict diffusion and are the most important features in the feature space. α is the least important feature as it is selected only 33.33% of the time. Referring to Table 6.4, ϕ and λ are selected together all of the time as compared to other combination of the features. On the other hand, ϕ, α and λ, α are selected only 33.33% of the times as compared to other feature combinations.

Table 6.3: Feature ranking based on selection rate using flexible neural tree

Feature	Selection Rate (%)	Rank
ϕ and λ	100	1
α	33.33	2

6.1.5 t-Test

A random experiment is performed in Section 6.1.4 using FNT. In order to confirm that the results are not a mere coincidence, t-test is performed on different combination of feature

Table 6.4: 2-feature subset selection using flexible neural trees

Features	**Selection Rate (%)**
ϕ, λ	100
ϕ, α	33.33
λ, α	33.33

selection data as illustrated in the Table 6.5. In case of ϕ and λ t value is negative and the p-value is greater than 0.05. Therefore, the null hypothesis is accepted. There is statistically significant evidence that ϕ and λ belong to the same distribution. It implies that FNT naturally choosing ϕ and λ together all the times is not by chance and the results are actually correct. For the remaining cases, the t-value is high and p-value less than 0.05. This means that the features in each pair do not belong to the same distribution, thus, rejecting the null hypothesis.

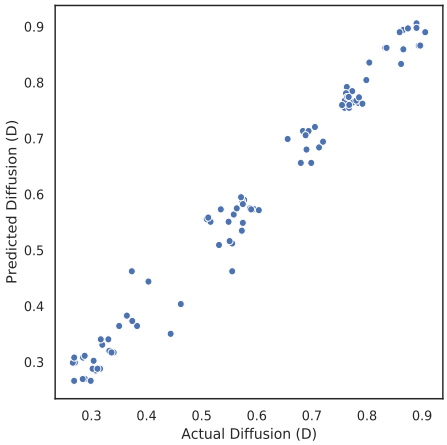
Table 6.5: t-test on FNT experiment data

Feature 1	**Feature 2**	**T value**	**p-value**
ϕ	λ	-1.28	0.21
ϕ	α	7.16	1.59×10^{-9}
λ	α	9.6	1.37×10^{-13}

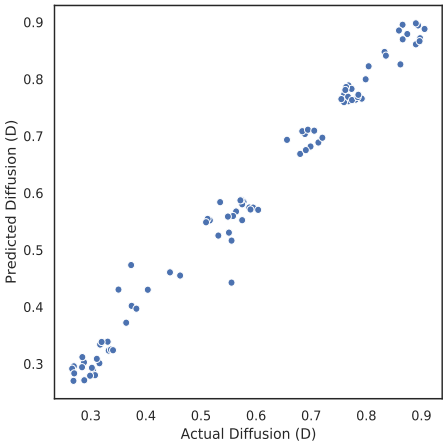
6.2 Predictive analysis

6.2.1 Regression

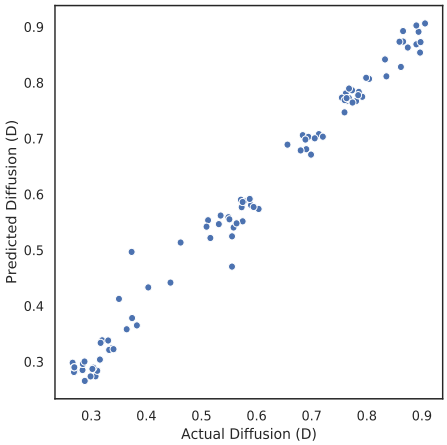
The results from feature analysis provide the best features subsets. Using these results, decision tree, random forest and extreme gradient boosting algorithms are trained with all possible combinations of 2 features and all the 3 features. Their R^2 scores are compared as shown in Table 6.6. In case of 3 features, random forest and extreme gradient boosting seem to learn the data set very well since they have the highest score. Figures 6.3(b),(c), 6.5 and 6.6 show that there are very few outliers (denoted by the red peaks) and the predicted diffusion values are almost the same as the actual values. Figure 6.4 show that there are slightly more outliers (denoted by the red peaks) as compared to random forest and extreme gradient boosting. However in case of 2 features the accuracy of DT and RF reduces but extreme gradient boosting remains unchanged. Furthermore, Figure 6.7 shows that the variance of the extreme gradient boosting reduces without α but the accuracy score remains unaffected by number of features. Interestingly, in case of 2 features, the accuracy scores of DT and RF reduce by 4%.



(a) Decision Tree



(b) Random Forest



(c) Extreme gradient boosting

Figure 6.3: Actual vs predicted diffusion

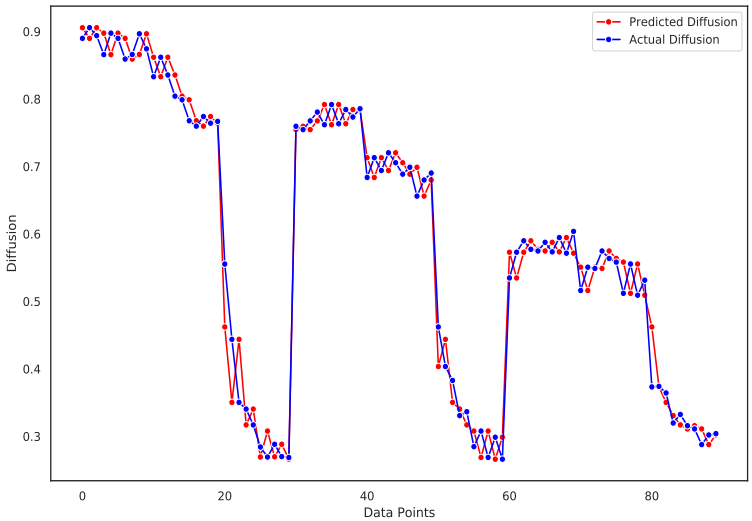


Figure 6.4: Decision Tree

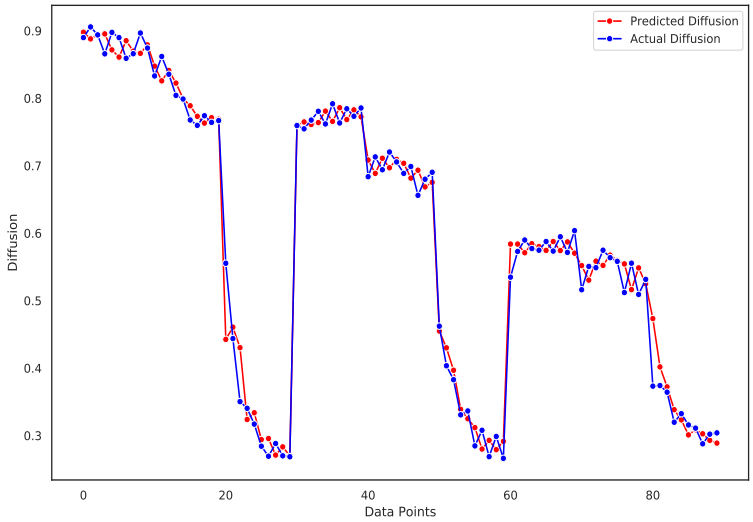


Figure 6.5: Random forest

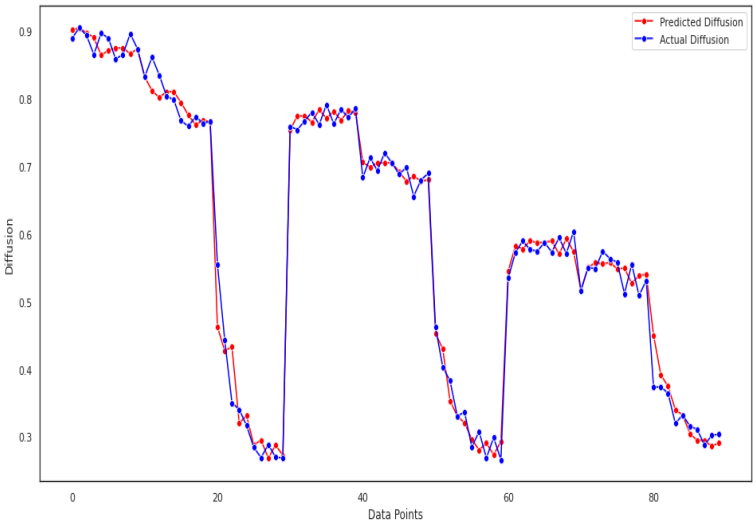


Figure 6.6: Extreme gradient boosting

Table 6.6: Comparison of R2 scores with repsect to the number of features

Algorithm	ϕ, λ	ϕ, λ, α
Decision Tree	0.94	0.97
Random Forest	0.94	0.98
Extreme Gradient Boosting	0.98	0.98

6.2.2 K-means and manual clustering

In order to apply k-means clustering to the data set, it is required to choose an appropriate k. This thesis uses Silhouette score to select the appropriate k. Silhouette score is computed for all the cluster sizes ranging from 1-10 as shown in Figure 6.8. It is evident that the Silhouette score is the least in case of $k = 6$. Following this, k-means algorithm is applied to the data set and it is divided into 6 clusters. However, it is observed that the clusters are formed based

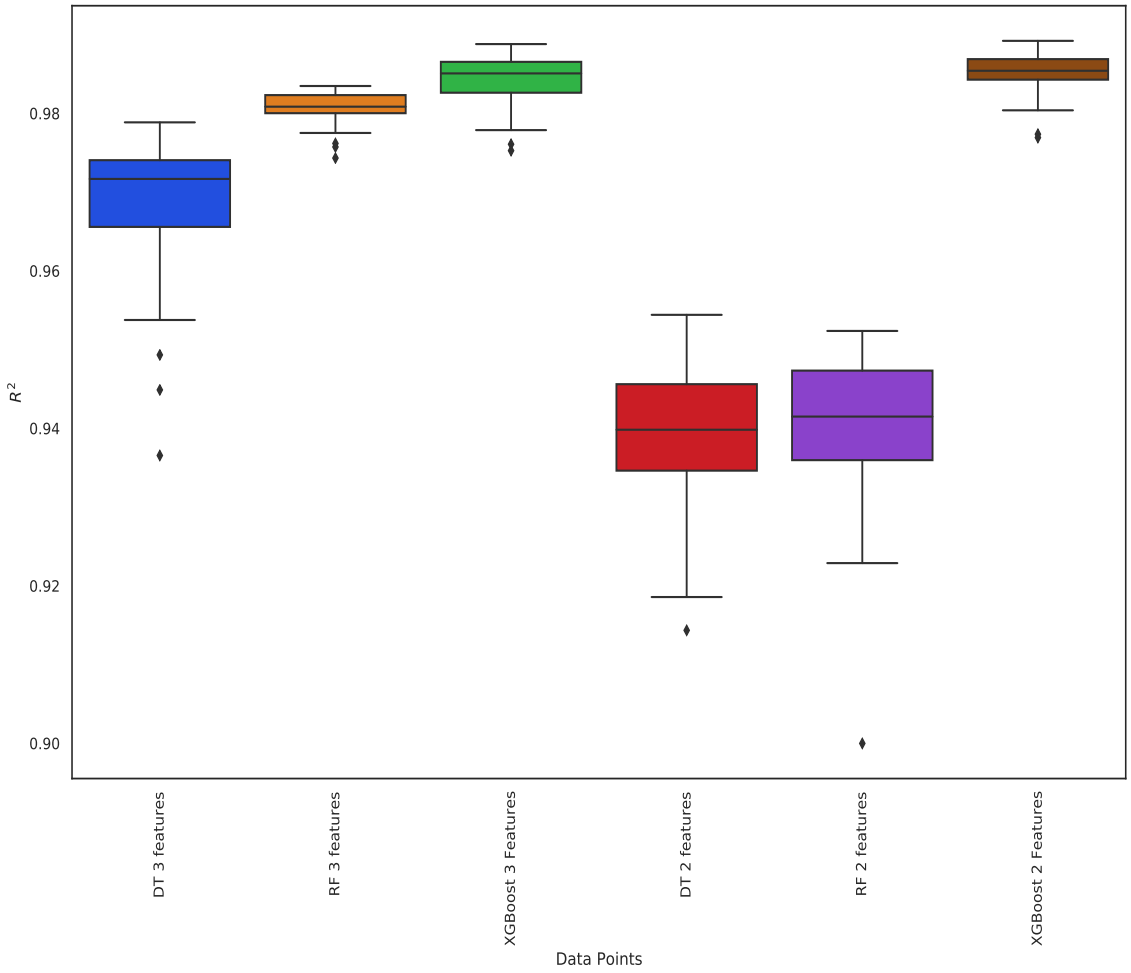


Figure 6.7: Comparison of R2 scores of random forest (RF), decision tree (DT) and XGBoost algorithms based on number of features

on α and not diffusion as shown in Figure 6.9. Moreover, the k seems to be very large and the clusters overlap even in case of α. This is not an expected outcome because the intention is to cluster the data set based on diffusion to extract some pattern associated with it.

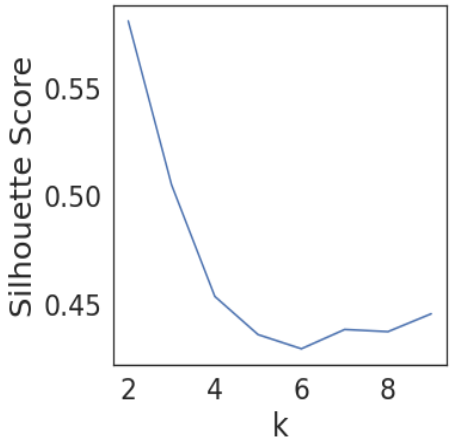
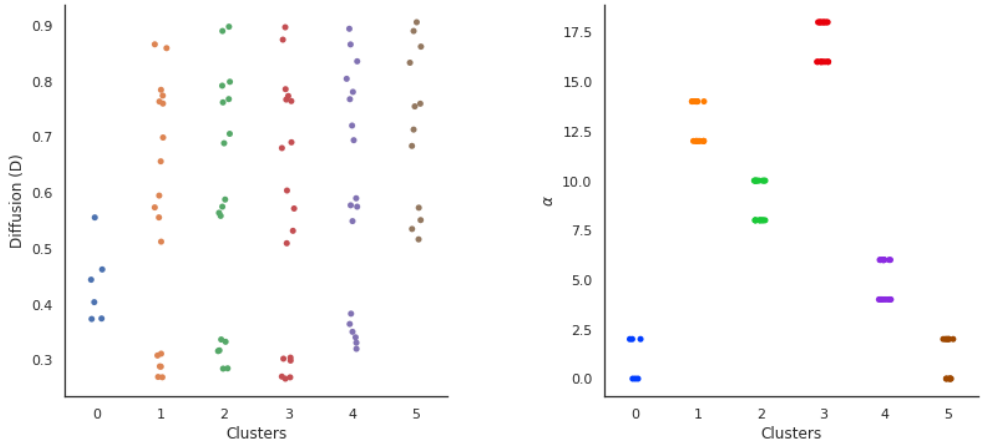


Figure 6.8: Silhouette score for k clusters



(a) Clusters vs. Diffusion (D) (b) Clusters vs. α

Figure 6.9: Behaviour of k-means with respect to α and D

To achieve the desired clusters, the data set is manually split into 2 clusters based on the histogram of diffusion as shown in Figure 6.10. It shows that the number of data points are almost equally distributed in case of $D = 0.6$ and using this information, the data is divided into 2 clusters; Class-1 corresponds to low diffusion ($D <= 0.6$) and Class-2 corresponds to high diffusion ($D > 0.6$). The manual clusters are plotted on the PCA grid in Figure 6.11 shows some overlap. This means that manually created clusters are clearly separated.

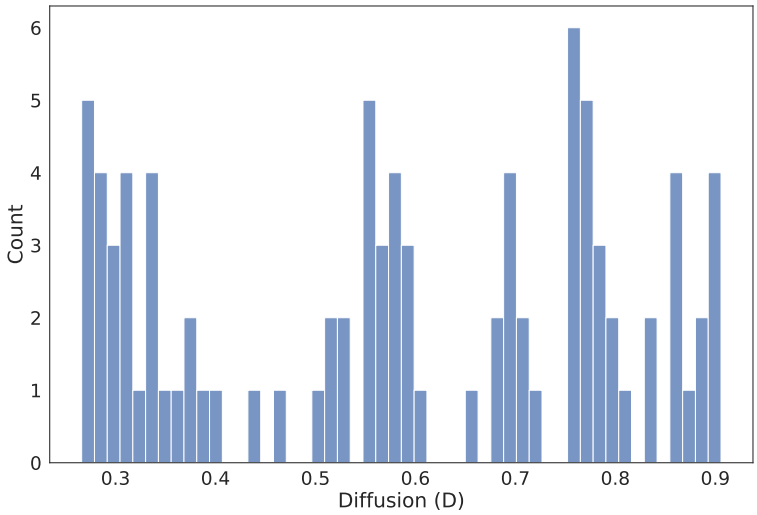


Figure 6.10: Histogram of diffusion

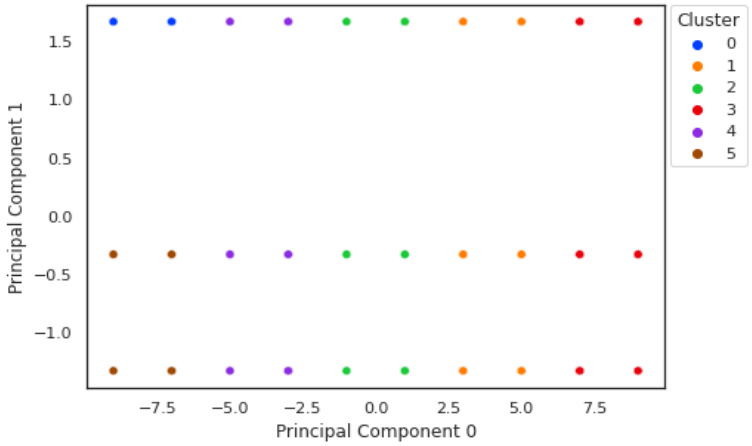


Figure 6.11: Manual clusters plotted on PCA grid

6.2.3 Classification

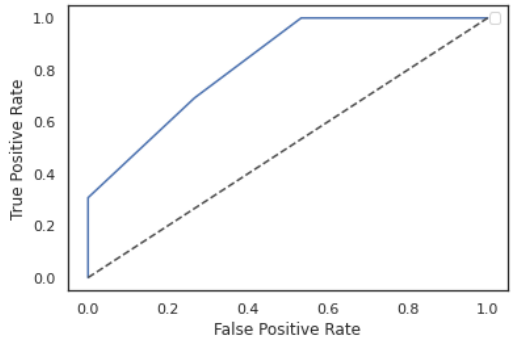
The original data set now also includes the discrete cluster labels generated in Section 6.2.2. Random forest (RF) and k-nearest neighbour (KNN) classification models are applied to the clustered data set. The size of Class-1 and Class-2 is 49 and 41 respectively. Here, the data set is balanced and performance measures such as precision, recall and ROC are used to evaluate the models. RF and KNN models are trained using 2 types of training. In Type-1 training, the models are trained on 70% of cluster-1 and 70% of cluster-2 and tested on the remaining data. In Type-2 training, the models are trained on 70% of cluster-1 and tested on the remaining data. This process is repeated 30 times and the average precision and recall scores are computed as shown in Tables 6.7 and 6.8.

Table 6.7: Precision (P) and recall (R) scores of random forest

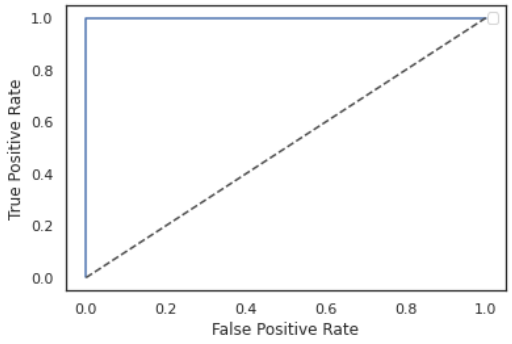
Features	Type of training	$D <= 0.6$		$D > 0.6$	
		P	R	P	R
ϕ, λ, α	Type 1	97.71	100	100	97.18
	Type 2	26.79	100	0	0
ϕ, λ	Type 1	97.92	99.78	99.76	97.44
	Type 2	26.79	100	0	0

Table 6.8: Precision (P) and recall (R) scores of k-nearest neighbour

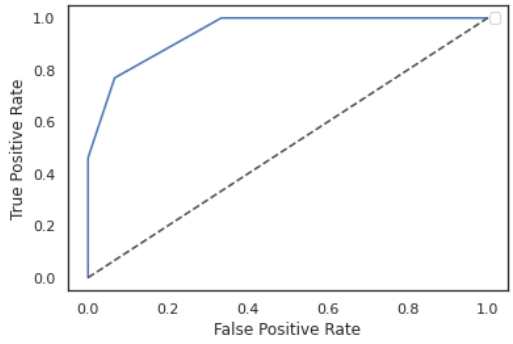
Features	Type of training	$D <= 0.6$		$D > 0.6$	
		P	R	P	R
ϕ, λ, α	Type 1	89	63.56	68.4	88.97
	Type 2	26.79	100	0	0
ϕ, λ	Type 1	97.61	73.11	77.19	97.18
	Type 2	26.79	100	0	0



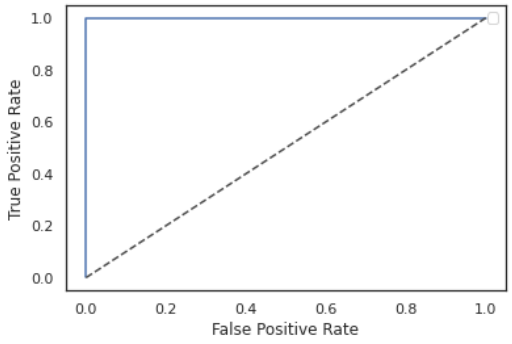
(a) KNN Type-2 training, 3 features



(b) RF Type-2 training, 3 features



(c) KNN Type-2 training, 2 features



(d) RF Type-2 training, 2 features

Figure 6.12: ROC curve of KNN and RF

It is observed that RF performs better than KNN in case of Type-1 training due to high precision and recall scores. This could be due to the ensemble method used by RF. The base learners in RF may predict the class label incorrectly but RF's output is the majority class voted by all the base learners rather than a single learner. Since it is rare for majority base learners to predict incorrect class label, RF is able to overcome bias. In case of Type-2 training, the results for both the models are same. This could be due to the fact that they are trained with data points containing only Class-1. Therefore, they are able to classify Class-1 correctly but are not able to recognise Class-2. Better results are obtained for 3 features as compared to 2 features. In case of random forest, removal of α does not improve the performance. On the other hand, the performance of the KNN improves when α is removed. In case of Type-2, the results remain same for RF and KNN. Figure 6.12 shows that in case of Type-2 training, 3 and 2 features, the ROC curve of KNN is closer to the diagonal line than that of RF. This means that RF performs better than KNN.

Chapter 7

Conclusions and Future Work

Descriptive and predictive ML algorithms are applied to both the diffusion data sets. In case of diffusion in polymer, PCA and EDA indicate that U_0 has the highest degree of influence on diffusion but the nature of influence is not clear. PCA and pearson correlation provide contradicting results in regards to the influence of a on D. Hence, the relationship between a and D is not clear. Finally, feature analysis shows that electrostatic potential (U_0), mesh size of the polymer matrix (a) and screening length (k) are the critical feature subsets. This is further confirmed by the reduction in variance of decision tree without ϵ and increase in its accuracy. However, random forest is the best predictive model with the least outliers because its accuracy and variance remains the same irrespective of ϵ. It is interesting to note that the models are unable to predict diffusion correctly with only U_0 and a. Therefore, U_0, a and k are required to predict diffusion with the highest accuracy. Since, k-means attempts to cluster the data points according to U_0, manual clustering is performed on the data set based on diffusion. However, when the manual clusters are plotted on the PCA grid, the clusters seem to overlap. This can be resolved by applying other clustering algorithms. In case of classification, random forest performs well with high precision and recall scores. To sum up, diffusion can be controlled effectively by choosing appropriate U_0, a and k values.

In case of ferrofluids, EDA, pearson coefficient and PCA indicate that as λ increases, diffusion decreases. However, no clear relationship can be established between ϕ, α and diffusion. According to PCA, λ and α seem to be the important features. On the contrary, feature analysis shows that ϕ and λ are important features. It is evident that λ is an important feature but it is not clear as to which of the two subsets denote critical features. Extreme gradient boosting learns the data set very well and its accuracy remains the same irrespective of number of features. Therefore, it is the best predictive model for ferrofluids to predict diffusion. Since, k-means attempts to cluster the data points according to α, manual clustering is performed on the data set based on diffusion. However, when the manual clusters are plotted on the PCA grid, the clusters seem to overlap. This can be resolved by applying other clustering algorithms. In case of classification, random forest performs well with high precision and recall scores.

Inspite of the results achieved, there still remains a wide scope for future work. New factors such as temperature can be added to the feature sets. The models can be trained to predict diffusion based on the new features along with the existing features. It has been difficult to anomaly detection model to detect anomalous diffusion. However, this can be conducted as part of further work. This thesis focuses on single objective to predict diffusion. However, using artificial neural network (ANN) it is possible to predict multiple targets especially in the case in the case of ferrofluids where, average cluster size and magnetization can also be predicted along with diffusion values.

Chapter 8

Reflection

It has been a challenging journey to work such cross domain project which is an amalgamation of molecular dynamics and machine learning. Since, I am new to both, I referred to a wide variety of literature such as books, journals, academic papers, website, videos to understand the concepts of ML and molecular dynamics (MD). Though some of the idiosyncrasies were confusing, I still managed to grasp the technical jargon of MD. Particularly, understanding the process of computer simulation was difficult. I learnt the importance of ML algorithms, model evaluation concepts such as repeated k-fold cross validation, performance measures and so on. I got the knack of approaching the solution to a real world problem. I understood the process of data mining to find the hidden patterns in the data which are otherwise invisible. My research helped me understand the working of different descriptive and predictive ML algorithms. Coding in python lead me to explore several libraries used in data analytics such as sklearn, seaborn, scipy, numpy and so on. It improved my python programming skills. While working on this thesis, I managed my timetable to attend the meetings with the collaborating professors. For each of the meetings, I had to present the results achieved in a way that a person who is novice to ML can comprehend. In this way, I learned how to translate the requirements in terms of ML and vice versa. I logged all the discussions, inputs and feedbacks to reprogram the course of my research. Thus, I developed management, communication and presentation skills. Overall, this thesis helped me to perceive the problem statement with the lens of a data scientist.

References

Allen, M. P. and Tildesley, D. J. (1989), *Computer Simulation of Liquids*, Clarendon Press, USA.

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, Springer.

Bisong, E. (2019), *Google Colaboratory*, Apress, Berkeley, CA, pp. 59–64.
URL: *https://doi.org/10.1007/978-1-4842-4470-8_7*

Brown, R. (1828), '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', *The Philosophical Magazine* **4**(21), 161–173.
URL: *https://doi.org/10.1080/14786442808674769*

Cakmak, U. M. and Cuhadaroglu, M. (2018), *Mastering numerical computing with NumPy*, Packt Publishing, Limited.

Callister, W. D. and Rethwisch, D. G. (2020), *Materials science and engineering*, 10 edn, Wiley.

Chen, J. C. and Kim, A. S. (2004), 'Brownian dynamics, molecular dynamics, and monte carlo modeling of colloidal systems', *Advances in Colloid and Interface Science* **112**(1), 159–173.
URL: *https://www.sciencedirect.com/science/article/pii/S0001868604000715*

Chen, S. and Yong, X. (2019), 'Janus Nanoparticles Enable Entropy-Driven Mixing of Bicomponent Hydrogels', *Langmuir* **35**(46), 14840–14848.

Chen, T. and Guestrin, C. (2016), Xgboost: A scalable tree boosting system, *in* 'Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '16, Association for Computing Machinery, New York, NY, USA, p. 785–794.
URL: *https://doi.org/10.1145/2939672.2939785*

Chen, Y., Yang, B., Dong, J. and Abraham, A. (2005), 'Time-series forecasting using flexible neural tree model', *Information Sciences* **174**(3), 219–235.
URL: *https://www.sciencedirect.com/science/article/pii/S0020025504002944*

Cover, T. and Hart, P. (1967), 'Nearest neighbor pattern classification', *IEEE Transactions on Information Theory* **13**(1), 21–27.

Cruz, C., Chinesta, F. and Régnier, G. (2012), 'Review on the Brownian Dynamics Simulation of Bead-Rod-Spring Models Encountered in Computational Rheology', *Archives of Computational Methods in Engineering* **19**(2), 227–259.

Desai, S. (2021), 'Msc dissertation', GitLab [Online].
URL: *https://gitlab.act.reading.ac.uk/aj842475/dissertation/*

Einstein, A. (1905), 'Über die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen', *Annalen der physik* **4**.

Fatin-Rouge, N., Starchev, K. and Buffle, J. (2004), 'Size effects on diffusion processes within agarose gels', *Biophysical Journal* **86**(5), 2710–2719.
URL: *https://www.sciencedirect.com/science/article/pii/S0006349504743258*

Friedman, J. H. (2001), 'Greedy function approximation: A gradient boosting machine.', *The Annals of Statistics* **29**(5).

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York.

Gordon, A. D., Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984), 'Classification and regression trees.', *Biometrics* **40**(3), 874.

Granik, N., Weiss, L. E., Nehme, E., Levin, M., Chein, M., Perlson, E., Roichman, Y. and Shechtman, Y. (2019), 'Single-particle diffusion characterization by deep learning', *Biophysical Journal* **117**(2), 185–192.
URL: *https://www.sciencedirect.com/science/article/pii/S0006349519305041*

Han, J., Kamber, M. and Pei, J. (2012), *Data mining*, Elsevier/Morgan Kaufmann.

Ilg, P. and Kröger, M. (2005), 'Anisotropic self-diffusion in ferrofluids studied via brownian dynamics simulations', *Physical review. E, Statistical, nonlinear, and soft matter physics* **72**(3 Pt 1), 031504–031504.

Iversen, G. R. and Gergen, M. M. (1997), *Statistics*, Springer New York.

Kalia, S., Kango, S., Kumar, A., Haldorai, Y., Kumari, B. and Kumar, R. (2014), 'Magnetic polymer nanocomposites for environmental and biomedical applications', *Colloid and Polymer Science* **292**(9), 2025–2052.

Karhunen, K. (1946), 'Zur spektraltheorie stochastischer prozesse', *Ann. Acad. Sci. Fennicae, AI* **34**.

Kelkar, K. M. and Bakal, J. W. (2020), 'Hyper parameter tuning of random forest algorithm for affective learning system', *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020* (Icssit), 1192–1195.

Li, S., Meng Lin, M., Toprak, M. S., Kim, D. K. and Muhammed, M. (2010), 'Nanocomposites of polymer and inorganic nanoparticles for optical and magnetic applications', *Nano Reviews* **1**(1), 5214–19.

Lieleg, O., Baumgärtel, R. M. and Bausch, A. R. (2009), 'Selective filtering of particles by the extracellular matrix: An electrostatic bandpass', *Biophysical Journal* **97**(6), 1569–1577.

Liu, Z., Liu, J., Cui, X., Wang, X., Zhang, L. and Tang, P. (2020), 'Recent Advances on Magnetic Sensitive Hydrogels in Tissue Engineering', *Frontiers in Chemistry* **8**, 2536–17.

Lloyd, S. (1982), 'Least squares quantization in pcm', *IEEE transactions on information theory* **28**(2), 129–137.

Loève, M. (1948), 'Processes stochastiques et mouvement brownien, ed', *P. Lévy (Paris: Hermann)* .

Lopez-Lopez, M. T., Durán, J. D. G., Iskakova, L. Y. and Zubarev, A. Y. (2016), 'Mechanics of magnetopolymer composites: A review', *Journal of Nanofluids* **5**, 479–495.

MacQueen, J. et al. (1967), Some methods for classification and analysis of multivariate observations, *in* 'Proceedings of the fifth Berkeley symposium on mathematical statistics and probability', Vol. 1, Oakland, CA, USA, pp. 281–297.

McKinney, W. (2013), *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*, 1 edn, O'Reilly Media.
URL: *http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1449319793*

McKinney, W. (2018), *Python for data analysis*, O'Reilly.

Megariotis, G., Vogiatzis, G. G., Schneider, L., Müller, M. and Theodorou, D. N. (2016), 'Mesoscopic simulations of crosslinked polymer networks', *Journal of Physics: Conference Series* **738**(1).

Meyer, R. A. and Green, J. J. (2015), 'Biodegradable polymer iron oxide nanocomposites: the future of biocompatible magnetism', *Nanomedicine (Lond)* **10**(23), 3421–3425.

Muñoz-Gil, G., Garcia-March, M. A., Manzo, C., Martín-Guerrero, J. and Lewenstein, M. (2019), 'Single trajectory characterization via machine learning', *New Journal of Physics* **22**.

Nakatsu, R. T. (2021), 'An evaluation of four resampling methods used in machine learning classification', *IEEE Intelligent Systems* **36**(3), 51–57.

Ojha, V. (2016), 'Neural-tree-software', GitHub [Online].
URL: *https://github.com/vojha-code/Neural-Tree-Software*

Ojha, V. K., Abraham, A. and Snášel, V. (2017), 'Ensemble of heterogeneous flexible neural trees using multiobjective genetic programming', *Applied Soft Computing* **52**, 909–924.
URL: *https://www.sciencedirect.com/science/article/pii/S156849461630494X*

Ojha, V., Schiano, S., Wu, C.-Y., Snasel, V. and Abraham, A. (2018), 'Predictive modeling of die filling of the pharmaceutical granules using the flexible neural tree', *Neural Computing and Applications* **29**.

Pavlov, Y. L. (2019), 'Random forests', *Random Forests* pp. 1–122.

Pearson, K. (1895), 'Notes on regression and inheritance in the case of two parents proceedings of the royal society of london, 58, 240-242'.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011), 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research* **12**, 2825–2830.

Pethrick, R., Amornsakchai, T. and North, A. M. (2014), *Introduction to Molecular Motion in Polymers*, Whittles Publishing.

Ryzhkov, A., Melenev, P., Holm, C. and Raikher, Y. (2015), 'Coarse-grained molecular dynamics simulation of small ferrogel objects', *J. Magnet. Magnet. Mater.* **383**, 277–280.

Schapire, R. E. (2013), Explaining adaboost, *in* 'Empirical inference', Springer, pp. 37–52.

Sewall, W. (1921), 'Correlation and causation', *Journal of Agricultural Research* **7**(7), 557–585.

Student (1908), 'The probable error of a mean', *Biometrika* **6**(1), 1–25.
URL: *http://www.jstor.org/stable/2331554*

Tan, P.-N., Steinbach, M. and Kumar, V. (2016), *Introduction to data mining*, Pearson Education India.

Uddin, S., Khan, A., Hossain, M. E. and Moni, M. A. (2019), 'Comparing different supervised machine learning algorithms for disease prediction', *BMC Medical Informatics and Decision Making* **19**(1), 1–16.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P. and SciPy 1.0 Contributors (2020), 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python', *Nature Methods* **17**, 261–272.

Wang, Y., Li, Z., Ouyang, J. and Karniadakis, G. E. (2020), 'Controlled release of entrapped nanoparticles from thermoresponsive hydrogels with tunable network characteristics', *Soft Matter* **16**(20), 4756–4766.

Wang, Z., Holm, C. and Müller, H. W. (2002), 'Molecular dynamics study on the equilibrium magnetization properties and structure of ferrofluids', *Physical review. E, Statistical, nonlinear, and soft matter physics* **66**(2 Pt 1), 021405–021405.

Young, R. J. and Lovell, P. A. (2011), *Introduction to polymers*, Taylor Francis Inc.

Zhang, X., Hansing, J., Netz, R. R. and Derouchey, J. E. (2015), 'Particle transport through hydrogels is charge asymmetric', *Biophysical Journal* **108**(3), 530–539.
URL: *http://dx.doi.org/10.1016/j.bpj.2014.12.009*

Zhao, X., Kim, J., Cezar, C. A., Huebsch, N., Lee, Kangwon Bouhadir, K. and Mooney, D. J. (2011), 'Active scaffolds for on-demand drug and cell delivery', *PNAS* **108**(1), 67–72.

Zhou, H. and Chen, S. B. (2009), 'Brownian dynamics simulation of tracer diffusion in a cross-linked network', *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* **79**(2), 1–10.

Zhu, J., Wei, S., Chen, M., Gu, H., Rapole, S. B., Pallavkar, S., Ho, T. C., Hopper, J. and Guo, Z. (2013), 'Magnetic nanocomposites for environmental remediation', *Advanced Powder Technology* **24**(2), 459–467.

Appendix A

An Appendix Chapter

A.1 In a nutshell: Molecular Dynamics, Langevin Dynamics, Brownian Dynamics

The theory given in this section is found in Allen and Tildesley (1989). For a collection of N particles we have Newton's equation of motion given by Equation A.1.

$$m_i \frac{d}{dt} v_i = F_i^{tot} \tag{A.1}$$

where m_i and v_i is the mass and velocity of particle i, respectively, and F_i^{tot} denotes the total force acting on particle i and $i = 1, \ldots, N$. The velocity is related to the position r_i by $\frac{d}{dt} r_i = v_i$. For given forces $F_i^{tot}(r_1, \ldots, r_N, v)$, our task is to solve $6N$ first order ordinary differential equations to find the trajectories $r_1(t), \ldots, r_N(t)$ for given initial conditions. This is the program of **Molecular Dynamics**.

Now consider small particles (typically 1-100nm) moving through a viscous liquid which acts as a solvent. This problem can be solved using Equation (A.1) by including all solvent molecules and calculating the forces. However, this is terribly inefficient, since our particle barely moves, while the solvent molecules. Therefore, the idea is to neglect the solvent, use Equation. (A.1) only for our particles of interest and include the effect of the solvent implicitly ("implicit solvent"). To do this, the forces are split into $F_i^{tot} = F_i + F_i^{fric} + F_i^B$, where $F_i = -\nabla_i U$ are potential forces, $F_i^{fric} = -\xi v_i$ is the friction force that particle i experiences when travelling though a viscous liquid with friction coefficient ξ. For spherical particles $\xi = 3\pi\eta\sigma$ with η the viscosity of the solvent, σ the hydrodynamic diameter. Finally, $^{\mathrm{B}}_{i}$ are Brownian ("random") forces. Plugging these into Equation (A.1) gives us the **Langevin** equation

$$m_i \frac{d}{dt} v_i = F_i - \xi v_i + F_i^B \tag{A.2}$$

While Equation (A.2) is of the same form as Equation (A.1), it is important to emphasize that all particles and all their interactions are included in Equation (A.1), while the Langevin equation does not consider the solvent explicitly and therefore includes friction and random forces.

The fastest process in the Langevin equation is the inertial relaxation, i.e. the velocities attain their instantaneous stationary values. This happens on the inertial time scale $\tau_I = m/\xi$ (can e.g. be seen from Equation (A.2) for $F_i = 0$). In many cases of interest, τ_i is very short (typically $\tau_I \approx 10^{-8} \ldots 10^{-6}$ s) and therefore we are looking to a further simplification: set the particle acceleration to zero ("overdamped motion"). Then Equation (A.2) becomes a

first-order differential equation that we can write (using $v_i = \frac{d}{dt} r_i$) as Equation A.3, which is the equation of **Brownian Dynamics**.

$$\frac{d}{dt} r_i = \frac{1}{\xi} F_i + \frac{1}{\xi} F_i^B \tag{A.3}$$