Predicting the Molecular Weight Distribution of Plastics from their Flow Data

Benjamin Jeffrey
201479413

Supervised by Professor Daniel Read

Submitted in accordance with the requirements for the
module MATH5872M: Dissertation in Data Science and Analytics
as part of the degree of

Master of Science in Data Science and Analytics

The University of Leeds, School of Mathematics

November 2021

The candidate confirms that the work submitted is his/her own and that appropriate credit has been given where reference has been made to the work of others.

Abstract

Plastics are becoming increasingly abundant in our society. With ever-increasing production, managing the disposal of plastics is as important as ever. Ideally, we would want to recycle more plastics in order to prevent used plastic material from ending up in landfills or the environment. To do so, a quick characterisation of plastics is essential, as there is a wide variety of plastics with differing physical properties. An important characteristic of plastics that can tell us a lot about their physical properties is their molecular weight distribution. Typically, the molecular weight distribution is determined using gel permeation chromatography, which is complex and requires specialist equipment not available to every plastics recycler. A possible alternative to this approach is using a plastic's flow properties derived from oscillatory shear experiments to predict its molecular weight distribution. In this project, we investigate whether this approach is feasible using neural networks to predict the distributions. We show that in the bimodal case, where the distribution has only one peak, this approach is viable, and we are able to predict the distribution parameters with low errors. In the case where the distribution has two peaks, the results are less conclusive, and the feasibility of using this method depends on the error tolerance of the user.

Acknowledgements

I would like to thank Professor Daniel Read for supervising this dissertation and being very helpful with any questions I had throughout the project.

Furthermore, I would like to thank Dr Chinmay Das for providing valuable guidance on using the software used for generating the datasets for this project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Plastics have been used by humans in one form or another for a long time, with discoveries of humans using natural rubber dating as far back as 1600 BC (Hosler et al., 1999). During the 20th century, the production of synthetic plastics started to accelerate rapidly due to technological advancements and the synthesis of new classes of polymers (Andrady and Neal, 2009). Thanks to their many beneficial properties, like their durability, chemical- and light resistance, being easily shaped into any desirable form, and low prices, plastic demand has continued to grow

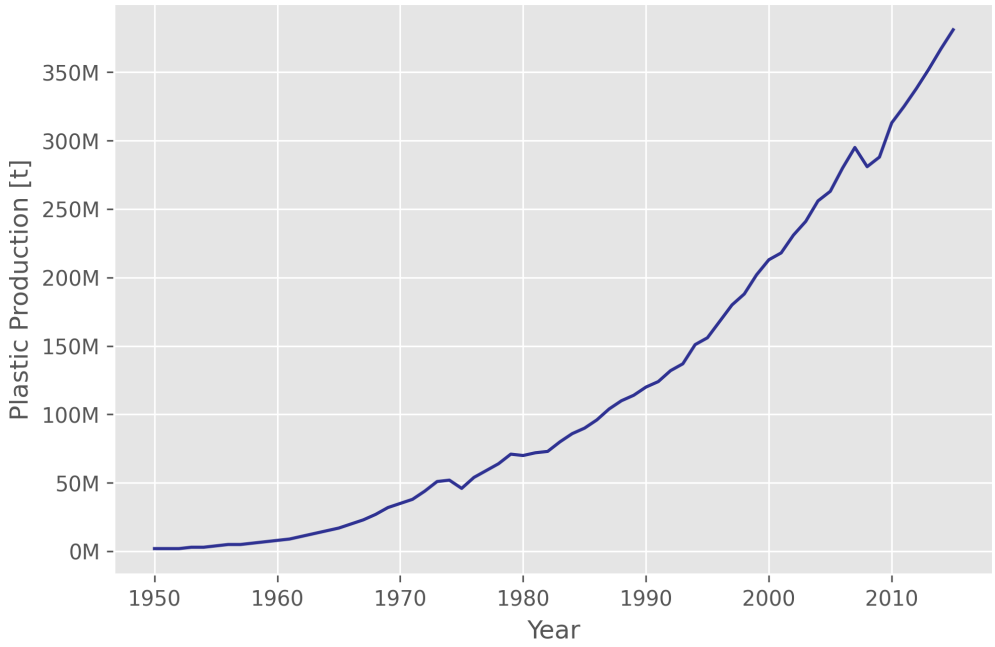

Figure 1.1: Annual global plastic production, 1950 to 2015 (Geyer et al., 2017)

year after year (Andrady and Neal, 2009). Figure 1.1 (Geyer et al., 2017) shows the global annual plastic production from 1950 to 2015, illustrating the growing abundance of plastics.

Most plastics do not degrade over time, which introduces the problem of what happens to them after they have served their purpose (Ali et al., 2021). There are currently three options for the disposal of plastics that are commonly employed; discarding the plastics in landfills or the environment, incinerating them, or recycling. Figure 1.2 (Geyer et al., 2017) shows the estimated share of each disposal method from 1980 to 2015. Notably, both incineration and recycling are gaining importance and will soon be the main disposal methods for plastics if the current trends continue.

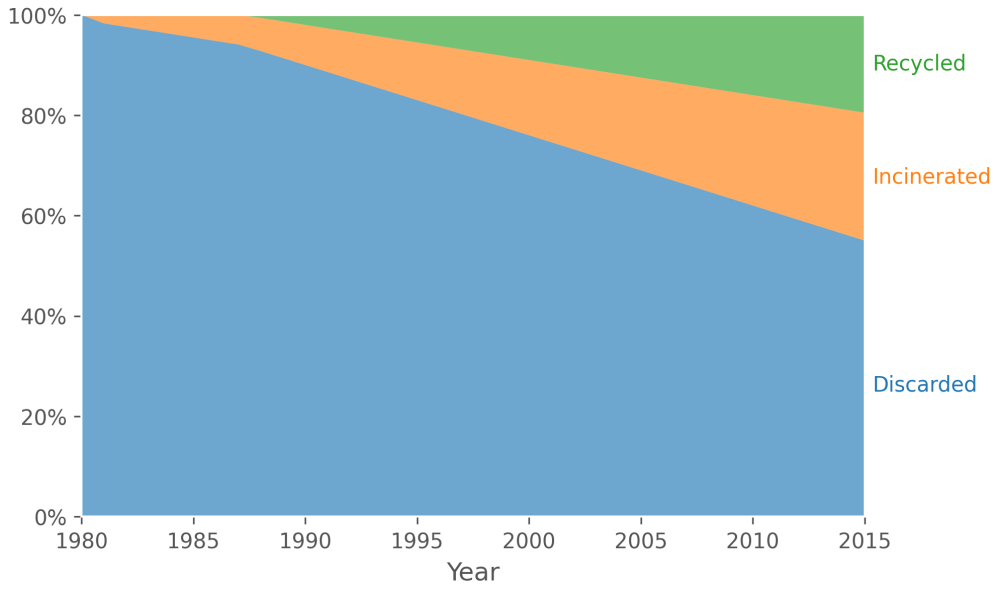

Figure 1.2: Estimated share of global plastic waste that is discarded, incinerated or recycled (Geyer et al., 2017)

Discarding plastics has many adverse effects on the environment, such as soil infertility from landfills and pollution through improper disposal (Ali et al., 2021). Eriksen et al. (2014) estimated that plastic pollution resulted in at least 5.25 trillion plastic particles being afloat at sea, which has adverse effects on marine ecosystems. The incineration of plastics also has its drawbacks, mainly the emission of greenhouse gases contributing to global warming (Ali et al., 2021).

Which leaves recycling, which also has drawbacks and problems that have not yet fully been solved. Most types of plastics cannot be reused for the same application and are instead used for different purposes, which often results in a downgrade of the plastic. Due to this, the material often changes with each iteration of a recycling loop. Sorting and identifying future use cases for the wide range of plastics is, therefore, one of the main challenges of recycling (Hopewell et al., 2009). Even when the type of plastic is known, its molecular structure can influence the

material's mechanical properties and the applications it can be used for after recycling. A quick way of characterising plastics is therefore invaluable for improving the recycling process.

One property of plastics that significantly impacts the material's mechanical attributes is the molecular weight of the polymers that make up the material (Jansen, 2016). The manufacturing process of polymers introduces variance into the molecular weight of the polymers as we currently cannot synthesise identical polymers consistently. Plastics, therefore, have a molecular weight distribution (MWD) that influences their mechanical properties (Balani et al., 2014).

The primary process used for determining the MWD of plastics in the industry is gel permeation chromatography (GPC). However, this approach is not always successful as it requires polymers to dissolve in a solvent, which is not always possible (Dealy et al., 2018). GPC is also costly and requires specialist equipment typically only available to larger laboratories. Sending off samples to external laboratories is slow, making it an unideal solution for plastic recyclers to rapidly identify batches of recycled materials. A possible alternative to characterising plastics and deducing their MWD is to use rheological data describing the material's flow properties.

This project aims to investigate whether using statistical modelling to predict a plastic's MWD from its flow data is a feasible alternative to GPC.

1.2 Description of Work

In this dissertation, we used statistical models to predict the MWD of a given plastic from its flow data. To do so, we first collected rheological data for our chosen plastic, polystyrene. As we lacked the equipment to perform the oscillatory shear experiments required to gather the flow data and collecting a sufficiently large dataset using these experiments would have been very time-intensive, we instead used software to generate our datasets. MWDs of plastics can often be bimodal, which led us to create separate datasets to cover this case in addition to the dataset generated from unimodal MWDs. We then explored the dataset's target and feature attributes and verified the quality of the data. The collection and exploration of the data are covered in Chapter 3.

Subsequently, we built and trained neural networks for the unimodal and bimodal cases using the collected datasets. The network architecture was optimised to minimise the mean relative error of the model predictions. The modelling and hyperparameter tuning are discussed in Chapter 4.

In order to determine if a statistical modelling approach using flow data is a feasible alternative to GPC, we analysed this method's performance in different aspects. We investigated the model performance using datasets of varying size to gauge the effect of additional data and find the point of diminishing returns for generating more data. Furthermore, we restricted the flow data

to various frequency ranges to see how prediction performance varies among them. We also investigated model performance in different regions of the target attribute space. In the bimodal case, we examined how the prediction performance changes if we require the two peaks of the distribution to be various minimum distances apart and if we require a minimum percentage of each component in the compound. The analysis of the statistical modelling approach is covered in Chapter 5.

Chapter 2

Background

This chapter provides some background on concepts used in this dissertation and aims to establish the needed understanding of plastics, their molecular structure and rheology to follow the project.

2.1 Plastics and their Molecular Structure

Plastics are a type of material that can occur both naturally or be produced synthetically. Most people will be familiar with the concept of synthetic plastics through their abundance in everything from food packaging to drain pipes. However, natural variants such as latex or cellulose also exist (Science History Institute, 2019). They have several beneficial properties that lead to their widespread use, such as their chemical resistance and their strength, while simultaneously being easily worked as hot melt (Andrady and Neal, 2009). The properties of a particular plastic can vary a lot and are determined by the molecular structure of the material (Jansen, 2016). In recycling and various other use cases, it is, therefore, desirable to gain insights into the material's molecular structure, which allows understanding its mechanical properties and working with the material more efficiently.

On a molecular level, plastics are polymers, which are long chains of repeating units. These units, called monomers, are molecular structures that bond together to form a polymer through a process called polymerisation (Science History Institute, 2019). An illustration of how polymers are formed from repeating monomers can be seen in Figure 2.1 (Generalic, 2018) using the example of polystyrene, the plastic we worked with throughout this project.

One way in which plastics can be differentiated is the monomers they are polymerised from. For instance, we can see in Figure 2.1 that styrene includes a ring of carbon and hydrogen atoms,

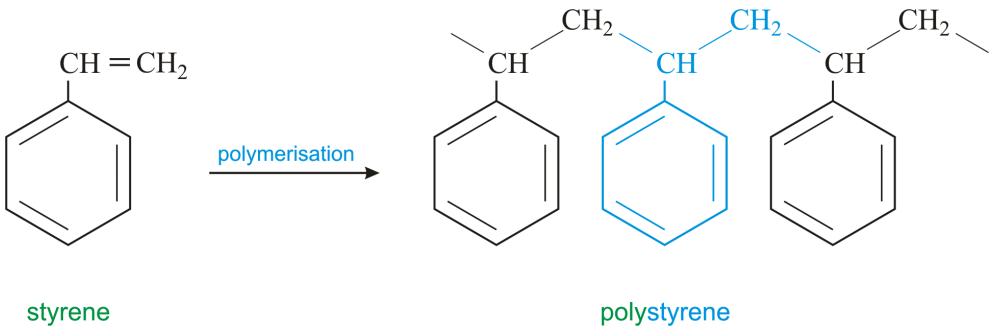

Figure 2.1: The polymerisation of styrene monomers resulting in polystyrene (Generalic, 2018)

called benzene. In contrast, ethylene, which is the monomer found in polyethylene, only has two carbon atoms in a double bond with four hydrogen atoms attached (Jansen, 2016).

Polymers can also vary in their degree of polymerisation, which is the number of repeating monomers in a polymer. This property is directly linked to the molecular weight of the polymer, which is calculated by multiplying the molecular weight of the repeating monomer by the degree of polymerisation (Balani et al., 2014). Furthermore, the monomers do not always form a straight chain but instead form side branches (Science History Institute, 2019).

In plastic melt, polymer chains are entangled and prevented from disentangling through intermolecular forces. An illustration of the entanglement of polymers can be seen on the left in Figure 2.2 (Impact Plastics, 2017). Generally, longer, higher molecular weight polymers will have a higher level of entanglement, which also influences the material's mechanical properties (Jansen, 2016). When the polymers are cooled and solidified, they can adopt different arrangements. Polymers that keep their chaotic, entangled form are called amorphous. Amorphous polymers often have lower melting points and tend to be transparent (Science History Institute, 2019). Conversely, polymers that form distinctive structures are called semicrystalline. Due to the large size of polymer molecules and their viscosity, polymers do not fully crystallise (Jansen, 2016). Semicrystalline polymers tend to be stronger and have higher melting points (Science History Institute, 2019).

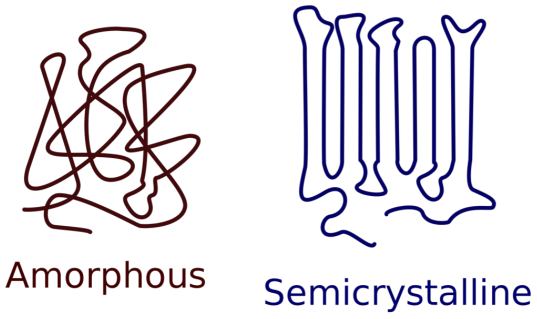

Figure 2.2: The entanglement of polymer chains in amorphous and semicrystalline polymers (Impact Plastics, 2017)

In practice, plastics will contain polymers with varying degrees of polymerisation due to variability in the synthesis of polymers. A compound will therefore have a molecular weight distribution (MWD) instead of a single value (Balani et al., 2014). We can calculate a statistical average from the distribution, of which two approaches are used in the industry.

The more straightforward method is using the **number average molecular weight** (M_n). Let n_i be the number of polymers having a specific molecular weight M_i from the set of all distinct molecular weights contained in the plastic. Then M_n is given by (Dealy et al., 2018, p. 18):

$$M_n = \frac{\sum n_i M_i}{\sum n_i}$$

The other option is to calculate the sum of all molecular weights in the compound multiplied by their weight fractions. This average is called **weight average molecular weight** (M_w) and is given by (Dealy et al., 2018, p. 19):

$$M_w = \frac{\sum n_i M_i^2}{\sum n_i M_i}$$

A plot of a possible MWD, showing the number-average and weight-average molecular weight, can be seen in Figure 2.3.

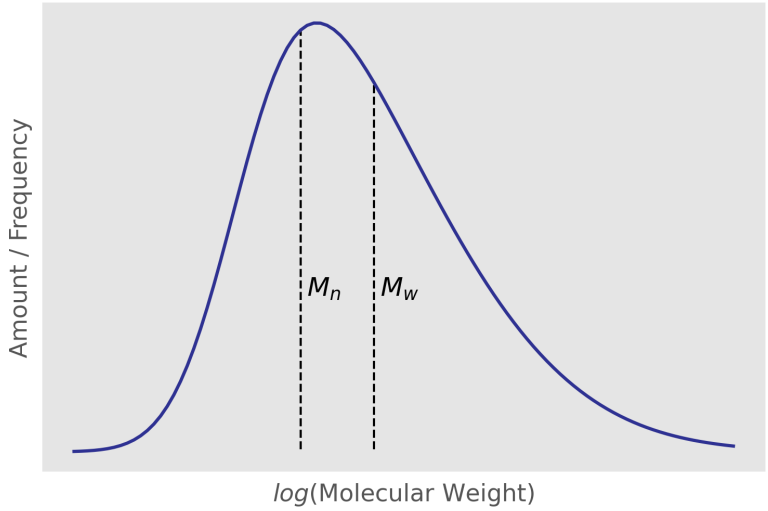

Figure 2.3: Molecular weight distribution showing the number average molecular weight (M_n) and weight average molecular weight (M_w)

The ratio of the weight average molecular weight to the number average molecular weight is called the **polydispersity index** (PDI) (Dealy et al., 2018, p. 19):

$$PDI = \frac{M_w}{M_n}$$

In the case where all polymers have precisely the same molecular weight (monodispersed poly-

mers), the PDI will equal 1. As we have already established, this is never the case with synthetic polymers outside of highly controlled lab experiments. In practice, PDI is usually larger than 1, and the higher it becomes, the wider the spread of molecular weights in the distribution (Dealy et al., 2018). Therefore, we can sufficiently describe the MWD using two of these parameters, for instance, using M_w and PDI.

In practice, the MWD will not always be a unimodal distribution, and instead, there may be two peaks in the distribution. In this bimodal case, the molecular weights of the polymer will be centred around two points, leading to there also being two averages (Dealy et al., 2018). We will denote averages of the lower molecular weight with a superscript s due to its shorter length polymers, e.g. M_w^s. Similarly, we will denote the higher weight averages with a superscript l due to its longer chain polymers, e.g. M_w^l. Figure 2.4 shows an example of a bimodal MWD, including M_w^s and M_w^l.

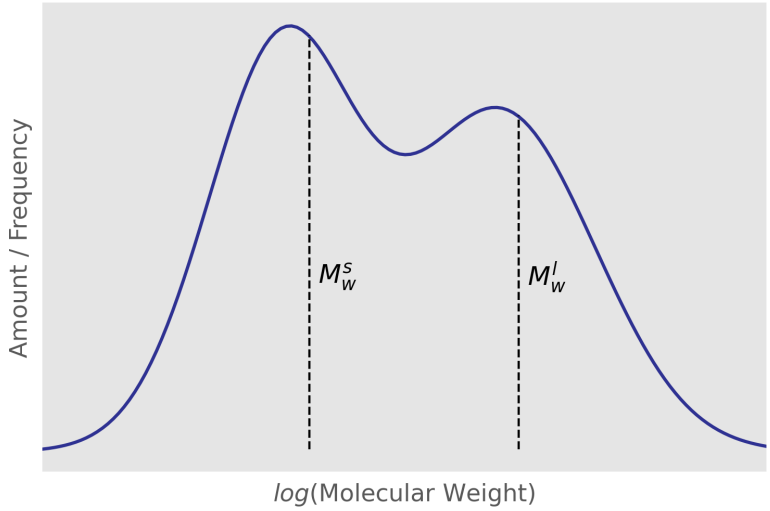

Figure 2.4: Bimodal molecular weight distribution showing the weight average molecular weight for the shorter polymers (M_w^s) and longer polymers (M_w^l)

Due to the two averages in the bimodal case, two dispersity measurements are also needed to describe the distribution, PDI^s and PDI^l. Higher modality distributions are possible, but they are not covered in this project.

We have previously discussed that the molecular weight of a polymer has a significant influence on its properties. Deducing the MWD of polymers is, therefore, the main focus of this project. More specifically, we aim to predict the M_w and PDI of given plastics from their flow data. In the bimodal case, this two-parameter problem is expanded to a five-parameter problem, which includes M_w^s, PDI^s, M_w^l, PDI^l and ϕ^l, where ϕ^l is the proportion of long-chain polymers in the compound. Technically we could add a sixth parameter, ϕ^s, to the problem definition, but this parameter can be trivially calculated from ϕ^l through $\phi^s = 1 - \phi^l$.

2.2 Rheology

Rheology is the study of the flow of matter. As such, it is focused on assessing the flow behaviour of materials. It is not restricted solely to liquids, as there are plenty of liquid-like materials between pure solids and pure liquids, especially when it comes to polymers (Anton Paar, 2021). Liquids are generally viscous, meaning they deform when subjected to small strains, such as shear stress, and do not return to their original form. The viscosity of a fluid is a measure of its flow resistance due to internal friction caused by molecules sliding over each other (Anton Paar, 2021). Therefore, a higher viscosity fluid, like honey, flows less quickly than a lower viscosity liquid, like water. Solids, on the other hand, are often linear elastic, meaning they do not deform permanently when subjected to minor strains and instead return to their original shape (Bower, 2009). Plastics are often viscoelastic, meaning they display behaviour characteristic of both viscous liquids and elastic solids (Anton Paar, 2021).

In order to characterise viscoelastic materials, the stress resulting from the material being subjected to sinusoidal deformations is recorded (Meyers and Chawla, 2008). This can be done through an oscillatory shear experiment, which is visualised in Figure 2.5 (Anton Paar, 2021). To perform the experiment, a sample of the material under inspection is clamped between two plates. The top plate is then moved back and forth at various frequencies, shearing the sample while the lower plate remains stationary (Anton Paar, 2021).

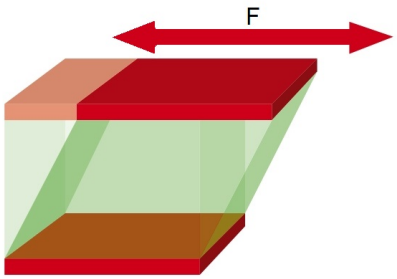

Figure 2.5: Oscillatory shear experiment (Anton Paar, 2021, Figure 8.2)

The oscillatory shear experiment produces the tested material's storage modulus (G') and loss modulus (G''). G' measures the energy stored in the sample when subjected to the shear stress. This tells us how elastic or solid-like the material behaves. Contrarily, G'' measures the energy lost when subjected to the shear stress. This tells us how viscous or liquid-like the material behaves (Meyers and Chawla, 2008).

The oscillatory shear experiment is typically performed at various frequencies in a given range. The resulting G' and G'' moduli are best visualised as curves over the frequency range, as can be seen in Figure 2.6.

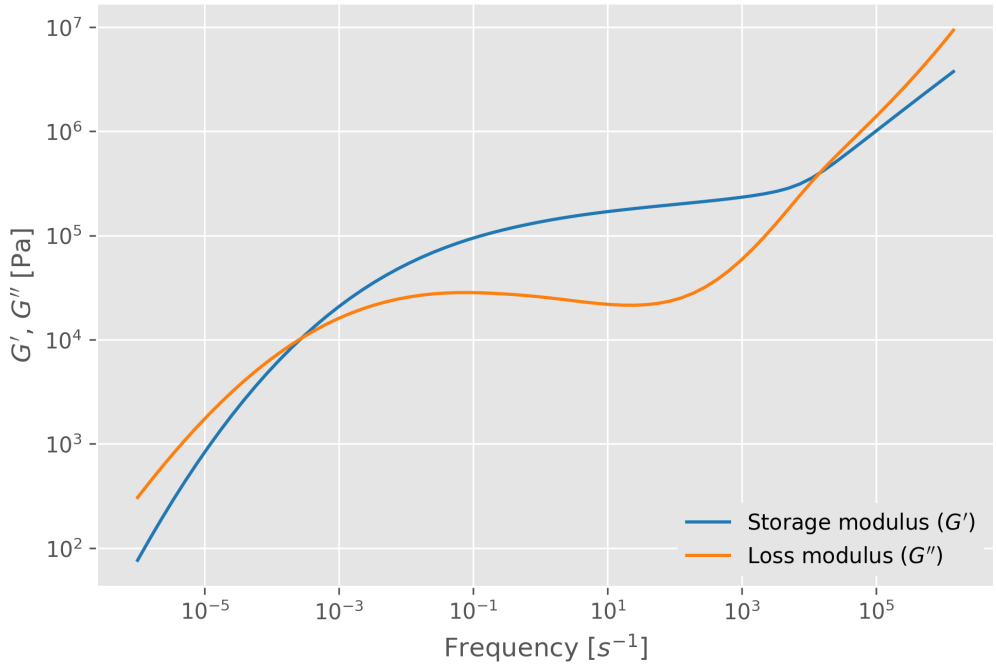

Figure 2.6: Storage modulus (G') and loss modulus (G'') over the frequency range of an oscillatory shear experiment

As we can see, the sample that was tested in this particular case had distinct viscoelastic behaviour. In the mid-frequency range, G' is higher than G'', meaning the material had predominantly elastic, solid-like characteristics at these frequencies. Contrastly, in the lower and higher range frequencies, G'' is higher than G', indicating the material shows more viscous, liquid-like properties at those frequencies.

The storage and loss modulus curves form the basis of the data used throughout this project. Each instance of our datasets includes information for these two curves across a frequency range. The aim is to use this flow data as features in a machine learning model to predict the MWD parameters covered in Section 2.1.

Chapter 3

Data

As discussed in Section 2.2, the data used in this project is rheological data in the form of storage modulus (G') and loss modulus (G'') curves. The oscillatory shear experiment that is usually used to collect this data requires specialised equipment we did not have access to. Furthermore, conducting the number of experiments required for a decently sized dataset would have been very time-intensive. We, therefore, opted to generate our datasets using software created by Das et al. (2006) for this project. This approach has several advantages. Firstly, it allowed us to control the target attribute space and generate samples with precisely the MWDs we desired. Furthermore, we were able to set the frequency range of G' and G'' as narrow or broad as we wanted. Lastly, using this method, we had the flexibility to generate more data when required and test the effect of using large datasets that could not feasibly have been collected by performing experiments.

The following sections cover the settings we used to collect our datasets, exploring the dataset's target attributes and features as well as verifying the quality of the data. This is done for both the unimodal MWDs as well as the bimodal MWDs.

3.1 Unimodal MWDs

3.1.1 Data Collection

The software we used to generate the flow data allows many adjustments to calculate precisely the G' and G'' curves we wanted. To generate an instance for the unimodal dataset, we had to specify the M_w and PDI we wanted to use. In order to give our models as much information as possible, we aimed to cover a sensible range for both these values as completely as possible. We, therefore, picked values at random from a uniform distribution of the target range. For the

M_w we picked values between 38,610 g/mol and 12,870,000 g/mol, which corresponds to 3 to 1000 times the entanglement molar mass (M_e) of polystyrene. For the PDI, we set the range to between 1.01 and 10. We could not use a lower bound of 1, as that would be a monodispersed polymer, which does not have a MWD. Finally, we also specified the frequencies at which we wanted the G' and G'' to be calculated. As we saw in Figure 2.6, the frequency is usually on a logarithmic scale. For our data, we started at 10^{-6} and multiplied by 1.5 every step up to 10^{6}. This resulted in 70 frequencies for which the software returned G' and G'' values, adding up to a total of 140 features for our dataset.

3.1.2 Data Exploration

In order to get a better understanding of the data we were working with, we wanted to explore the collected data points using statistical summaries and visualisations. The collected data set for the unimodal MWDs contains 150,000 instances. We looked at the target attributes first to ensure our data generation script ran correctly. Our goal was to generate the flow data from a uniform distribution of M_w and PDI values across their defined ranges. The scatter chart in Figure 3.1 shows the distribution of data points within the target attribute space. As we can see, the data points are evenly distributed and cover the entire possible range we set for the M_w and PDI attributes.

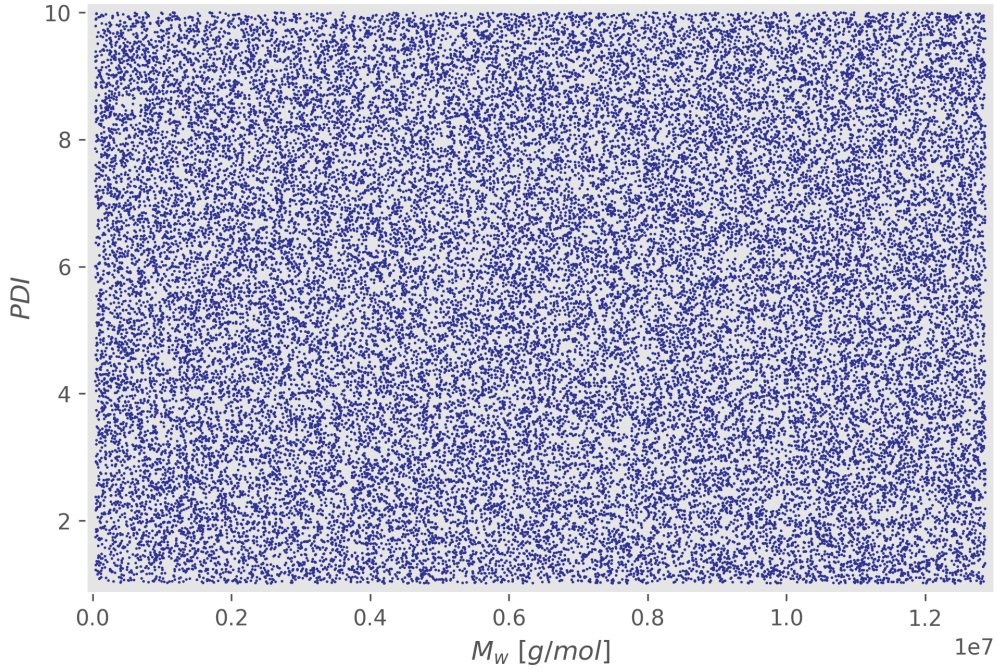

Figure 3.1: Scatter chart showing the distribution of instances in the target attribute space (sample of 40,000 instances, unimodal dataset)

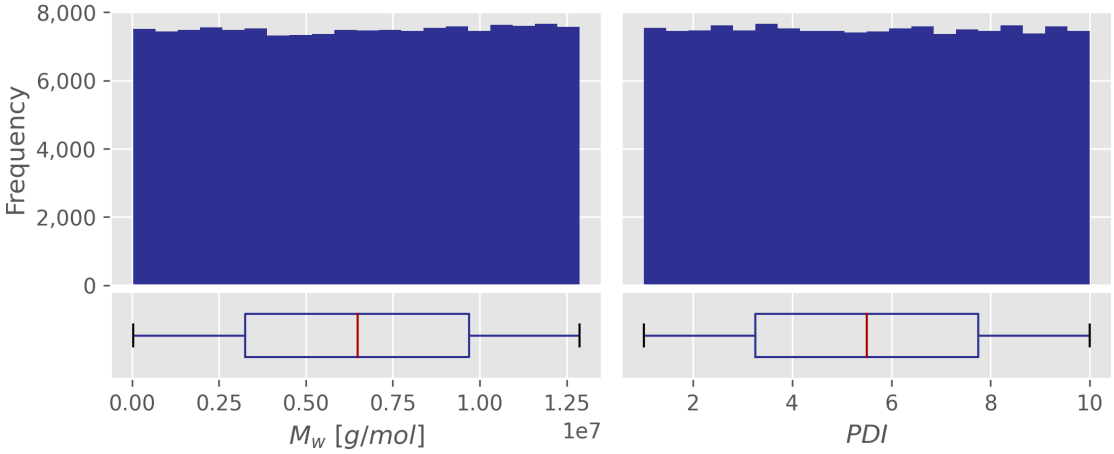

Figure 3.2: Histograms and boxplots of the M_w and PDI target attributes (unimodal dataset)

We checked this further by looking at Figure 3.2, which shows histograms and boxplots for the two target attributes, confirming they are uniformly distributed. The M_w attribute contains integer values, while the PDI attribute contains float values. Table 3.1 shows summary statistics of the attributes indicating their range and spread. As we can see, all values are in the ranges we specified. The M_w attribute contains almost all unique values, while every value of the PDI attribute is unique. This makes sense since the likelihood of picking the same integer twice when randomly generating is much higher than picking the same float value. There were no indications of irregularities or outliers in the data, and we did not find any missing values either.

Table 3.1: Summary statistics for the M_w and PDI target attributes, as well as the G' and G'' features (unimodal dataset)

	M_w [g/mol]	PDI	G' [Pa]	G'' [Pa]
count	150,000	150,000	10,500,000	10,500,000
unique count	149,146	150,000	5,365,653	5,246,056
mean	6,476,101	5.50	3.77×10^5	5.33×10^5
std	3,714,350	2.60	6.96×10^5	1.56×10^6
min	38,767	1.01	1.70×10^{-12}	1.63×10^{-4}
1st quartile	3,248,038	3.25	5.05×10^4	1.32×10^4
median	6,494,798	5.50	1.77×10^5	2.45×10^4
3rd quartile	9,701,439	7.75	2.46×10^5	7.00×10^4
max	12,869,978	10.00	3.75×10^6	9.55×10^6

Next, we explored the features of the dataset. As discussed in Section 3.1.1, the dataset contains 140 features, 70 for the G' and 70 for the G'' curve. All these values are float values. Summary statistics for the curves are also included in Table 3.1. One standout from these statistics was the considerably lower minimum value for both curves compared to the other statistics. Another was that roughly half the values are duplicate values. This can happen either when the curves have a perfectly horizontal section or if different curves pass through the same points.

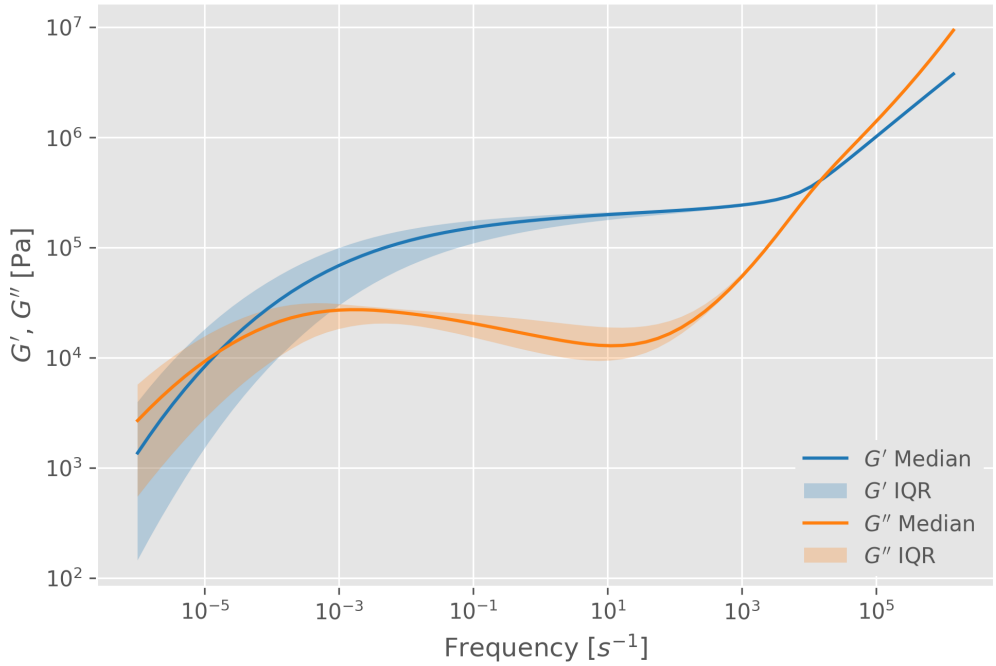

Figure 3.3: Median and IQR of the storage- and loss modulus curves (unimodal dataset)

Figure 3.3 helps visualise this by showing the spread of the features data. It shows the median and interquartile range (IQR) for G' and G'' across the entire frequency range. As we can see, the data is spread the widest in the lowest frequencies and still considerably large in the medium frequency range. However, in the higher frequencies, there is much less spread in the data. This was not unexpected, as the high frequencies are usually related to the fast motion over small distances of smaller parts of the molecule. These motions are less sensitive to variations of the MWD of the entire molecule as they are only influenced by the local structure.

We wanted to investigate the spread of the data in the low frequencies further by creating boxplots for the two features at the lowest frequency, which can be seen in Figure 3.4. For comparison, this chart also includes boxplots for a frequency in the high-frequency range, just before the median G' and G'' curves cross for the second time. The first thing of note in this chart was confirming the difference in spread, evident by the y-axis of the boxplots that have a far wider range for the low-frequency boxplots than the high-frequency ones. We can also see that the much smaller minimum values for both the G' and G'' curves are not anomalies. Though technically outliers using the $Q1 - 1.5 \times IQR$ rule, many other data points are similarly far from the median of the distribution. Therefore, there was no reason to doubt the validity of these data points, and we consequently did not remove them. It is also worth noting that the whisker's length depends on which scale is used to calculate them. Figure 3.4 was created by calculating the whiskers taking the log_{10} of the data due to the log scale we used throughout this project to visualise the storage and loss moduli. Had we used the data without this transformation, the whiskers would have extended to include all data points, and no outliers would have remained.

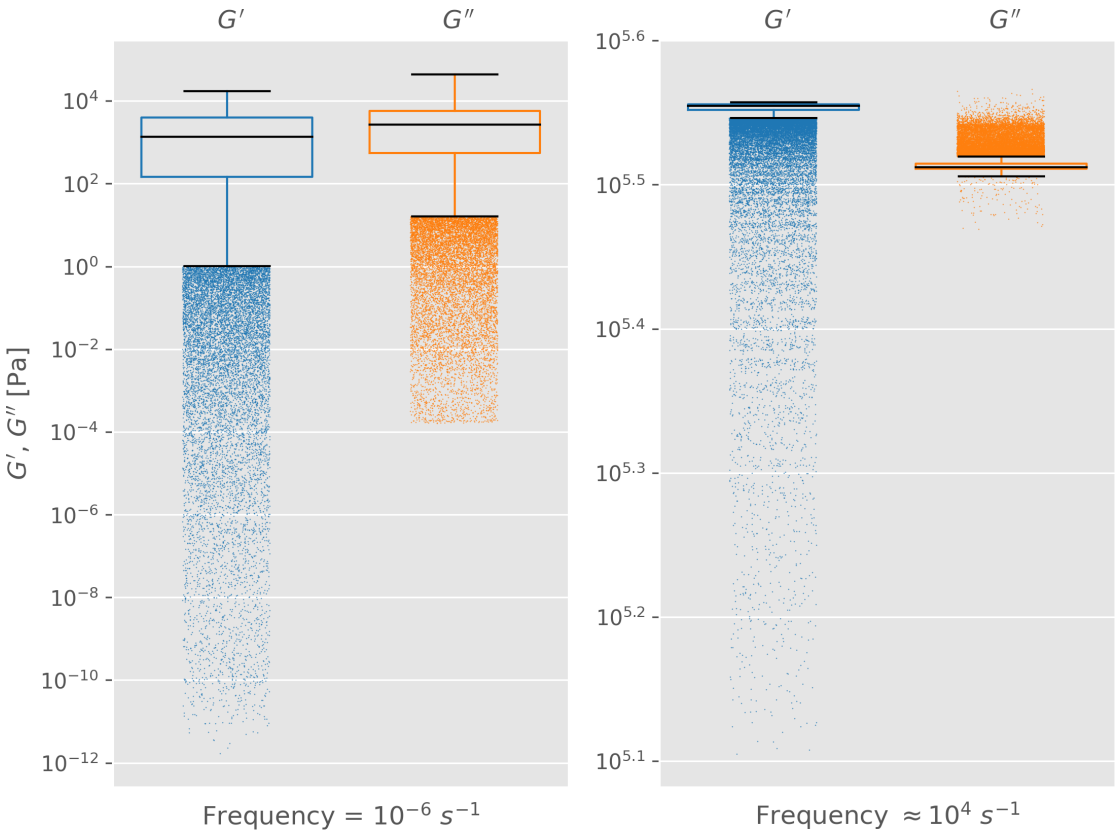

Figure 3.4: Boxplots showing the distribution of G' and G'' at the lowest frequency and at a high frequency (unimodal dataset)

One last thing we wanted to look at was the correlation between the features. The machine-learning models we choose are deliberately not susceptible to problems due to multicollinearity. However, we still wanted to investigate correlations to see if there was an interesting connection between features to be found. Figure 3.5 shows a correlation matrix for all the features in our dataset. There are four distinct sections because the features belong to either the G' or G'' curves. The top and left sections are connected to the storage modulus (G'), while the bottom and right sections are connected to the loss modulus (G''). Starting in the top left section, we see that the features of G' are mostly correlated to each other if they are close to each other. This is expected when dealing with smooth curves, of course. We also see this when looking at the correlation among the G'' features in the bottom right. However, unlike with the G' features, we also see negative correlations in the top right and bottom left corners. This indicates that a lower value in the higher frequencies usually accompanies a high value in the lower frequencies and vice versa. Lastly, the top right or bottom left shows the correlation between G' and G'' values. We can see that for lower frequencies of G'', we mostly have positive correlations, meaning higher values of G' usually accompany higher values of G''. For higher frequencies of G'', the opposite is true with a higher G' value usually appearing in tandem with lower values for G'' and vice versa.

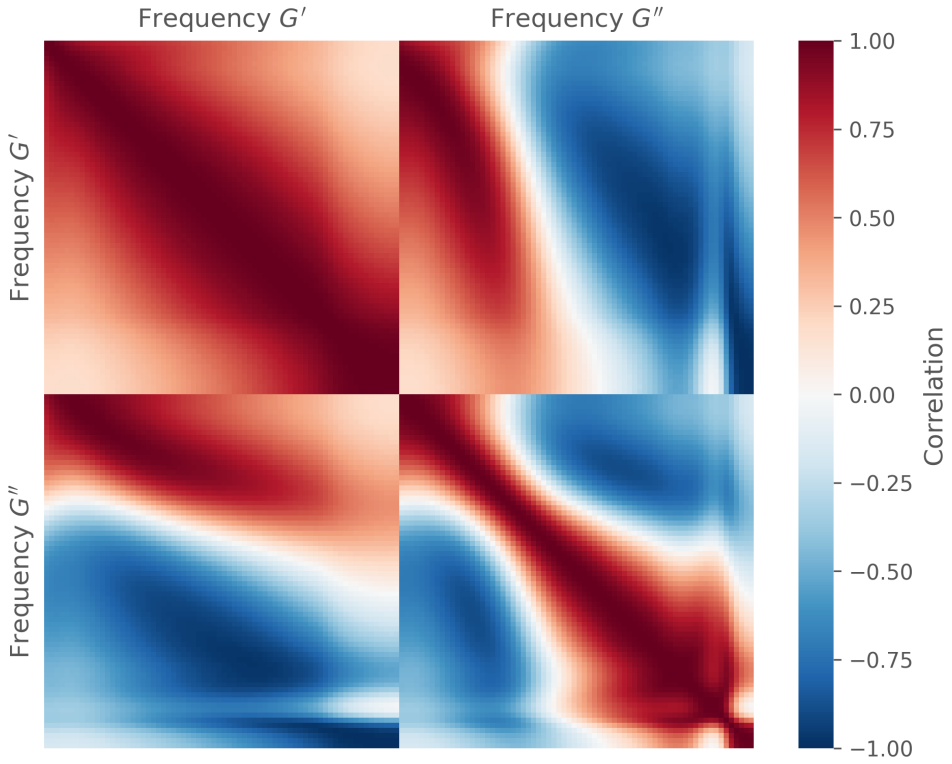

Figure 3.5: Correlation matrix of all features (unimodal dataset)

3.1.3 Verifying Data Quality

One of the benefits of generating our own data was that the data was inherently clean. As we saw in Section 3.1.2, there were no missing values, and all values remained in ranges we expected. We did not see any irregularities, outliers or unexpected values. Therefore, the quality of the data is primarily dependent on how accurate the software is we used for the data collection. However, this project focused on the feasibility of predicting the molecular weight distribution from flow data, not on building a ready to use prediction system. Such a system should be trained using at least a portion of data gathered through real oscillatory shear experiments for use in practice. The generated data worked well to gauge the feasibility of this approach, however, and it allowed us to gain many valuable insights, like the minimum amount of training samples needed for satisfactory results. We did not perform any feature selection for this project, as we wanted to give the models as much information to play with as possible. Furthermore, the neural networks we used can adjust the weights of features on input to effectively perform feature selection themselves. We did scale the features, however, as is covered in Section 4.2. This step is part of a data processing pipeline to feed data to the models and is, therefore, best discussed in the modelling section. Other than that, we did not manipulate the data any further and progressed to the modelling stage for unimodal MWDs using the data described in this section.

3.2 Bimodal MWDs

3.2.1 Data Collection

As in the unimodal case, we used the software created by Das et al. (2006) to generate the flow data we required for the bimodal predictors. The bimodal case is a five parameter problem, so we had to set all of these to generate an instance for our dataset. Like with the unimodal dataset, we first randomly choose a M_w and PDI in the same ranges as before. We then did this a second time for the second peak of the MWD. In Section 2.1 we characterized the different peaks by their weight. We, therefore, examined which of the two M_ws is smaller and set that and its corresponding PDI as M_w^s and PDI^s. The two remaining parameters were set as M_w^l and PDI^l. This was essential because the algorithms would have had a much harder time making meaningful predictions if the targets were not ordered. The final parameter we had to set was ϕ^l, the proportion of the high molecular weight component from the entire sample. This parameter we picked randomly between 0 and 1. We input these five parameters for each instance and received 70 features each for G' and G'' back, just like with the unimodal MWD. All other parameters, like the frequency range, were set to the same values used in the unimodal case.

There is one potential issue with bimodal MWDs that we have not addressed so far. That is the case where the peaks are so close together or spread so wide that they effectively form a unimodal distribution. When this happens, we as humans would not categorise such an instance as bimodal. Similarly, it could be challenging for the statistical model to predict the parameters reliably if the peaks are merged too much. To investigate this, we decided to test our models using datasets with variously strict restrictions on how close the peaks were allowed to be. The base (no restrictions) dataset was collected as described above. We also generated datasets where the M_w and PDI values had to fulfil the following restriction:

$$\frac{M_w^l}{M_w^s} > PDI_{max}^k, \qquad k \in \{1, 1.5, 2\}$$

PDI_{max} denotes the larger of the two randomly selected PDI values. We chose to use three different values for the exponent k to test how strict we had to separate the peaks for satisfactory results, with $k = 2$ being the most strict and the "no restrictions" option being the least strict. Due to this approach, we ended up with four datasets for the bimodal MWDs, though it should be noted that the more strict datasets contain the subset of the next less strict dataset that satisfies the restriction condition.

3.2.2 Data Exploration

As with the unimodal dataset, we first explored the collected data points using charts and statistical summaries. All four of the collected bimodal datasets contain 200,000 instances. Looking at the five target attributes allowed us to verify that the data generation ran as expected and to see the effect of the various restrictions between the datasets. The first thing we did, as in the unimodal case, was visualising the distribution of data points in the target attribute space, which can be seen in Figure 3.6 for the no restrictions bimodal dataset. The ϕ^l attribute is not shown here, as it is independent from the other attributes and is uniformly distributed in all datasets, as will become apparent in subsequent charts.

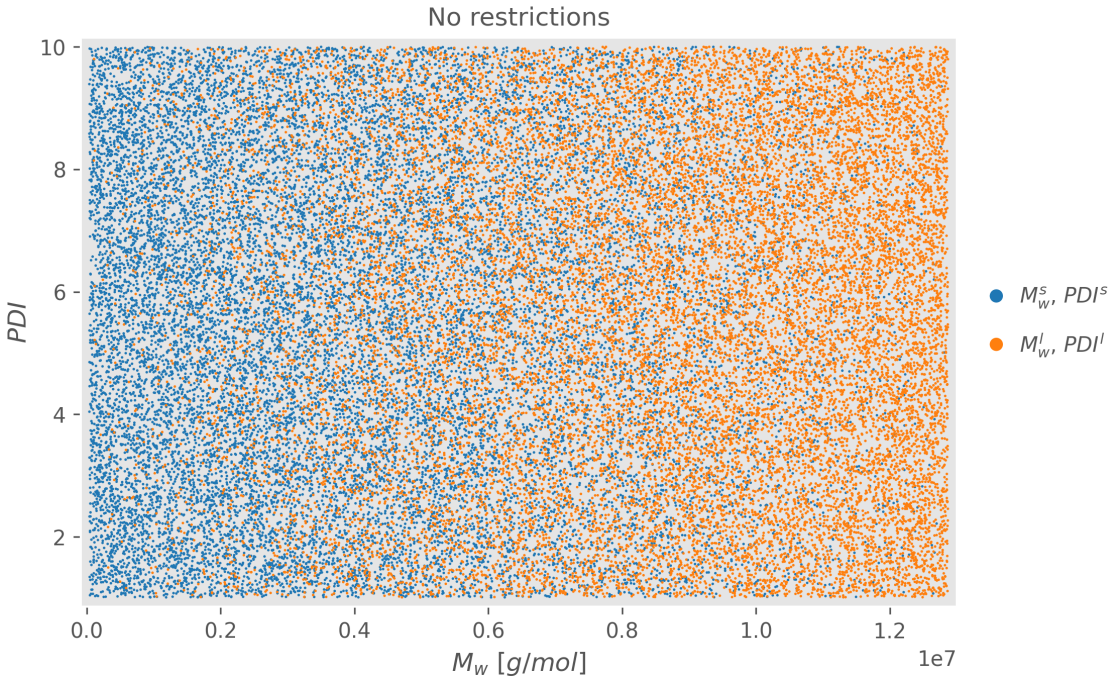

Figure 3.6: Scatter chart showing the distribution of instances in the target attribute space (sample of 20,000 instances, no restrictions bimodal dataset)

We can see that the data points are evenly distributed across the target attribute space and the defined ranges for M_w and PDI. Furthermore, the PDI values are evenly distributed among the higher molecular weight peaks (long peak) and the lower weight peaks (short peak). Unsurprisingly the long peak values dominate the higher molecular weight region on the right side of the chart, while the left side contains more short peak values.

Things get more interesting when examining the same chart for the $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ dataset, as shown in Figure 3.7. Here we can see that the data points are no longer evenly distributed in the target attribute space. Instead, the short peak values are concentrated towards the bottom left of the chart, while the long peak values are concentrated towards the lower right. This is due to the way the restriction is set up. With high PDI values, the molecular weight must be much further

apart, pushing the values to the right and left sides of the chart in the higher PDI regions. Towards the bottom of the chart, in the lower PDI regions, the values have more freedom for variations of the molecular weight while still satisfying the restriction condition.

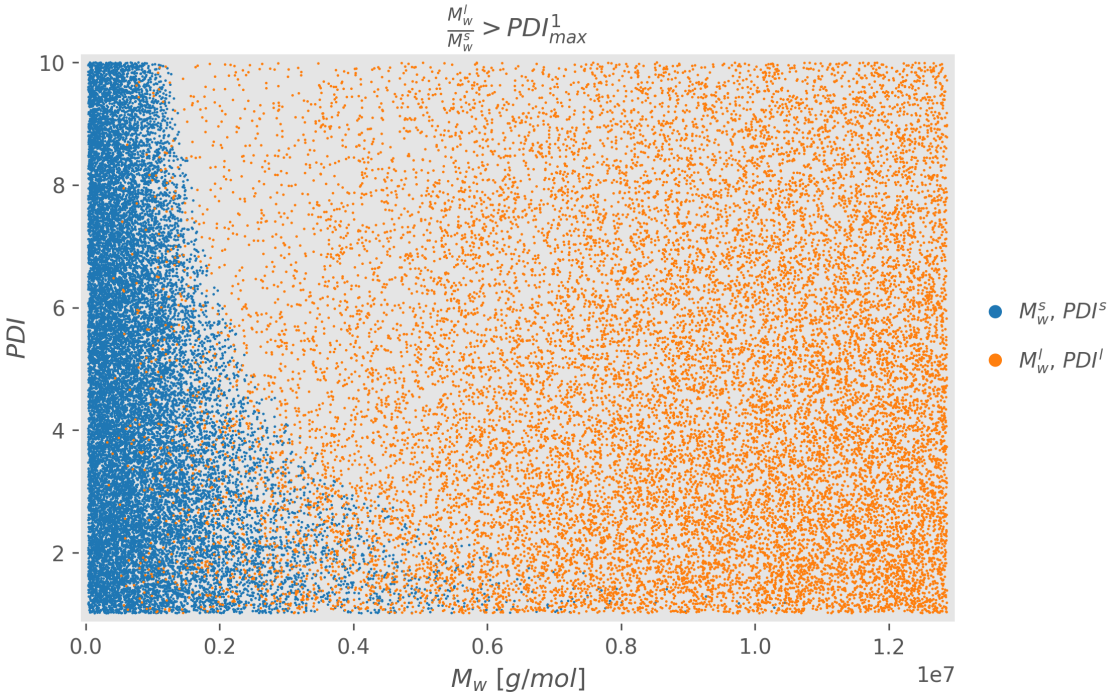

Figure 3.7: Scatter chart showing the distribution of instances in the target attribute space (sample of 20,000 instances, $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

This effect is more pronounced with the more restrictive bimodal datasets, as shown in Figure 3.8, where the data points are pushed to the lower left and right regions even more, especially the short peak values.

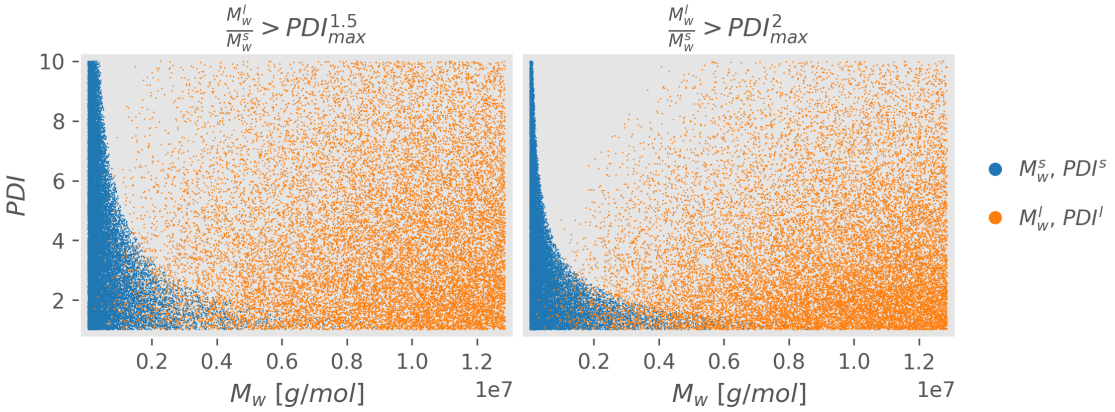

Figure 3.8: Scatter chart showing the distribution of instances in the target attribute space (sample of 20,000 instances, $\frac{M_w^l}{M_w^s} > PDI_{max}^{1.5}$ and $\frac{M_w^l}{M_w^s} > PDI_{max}^2$ bimodal datasets)

We can also see the difference the restrictions make when comparing the summary statistics, histograms and boxplots of the target attributes. Table 3.2 shows the summary statistics of the no restrictions bimodal dataset, while Table 3.3 shows the same statistics for the $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ dataset. As with the unimodal MWDs, the M_w attributes contain integer values while the remaining target attributes contain float values. The float attributes are again completely unique in all datasets, while the M_w attributes contain a few duplicates that are expected when sampling 200,000 integers randomly. The tables show some minor differences in the quartiles and mean of the PDI attributes and a substantial difference in most statistics for the M_w^s attribute.

Table 3.2: Summary statistics for the M_w^s, PDI^s, M_w^l, PDI^l, and ϕ^l target attributes (no restrictions bimodal dataset)

	M_w^s [g/mol]	PDI^s	M_w^l [g/mol]	PDI^l	ϕ^l
count	200,000	200,000	200,000	200,000	200,000
unique count	197,998	200,000	197,910	200,000	200,000
mean	4,316,553	5.51	8,608,313	5.50	0.500
std	3,029,765	2.60	3,027,457	2.60	0.289
min	38,640	1.01	59,881	1.01	0.000
1st quartile	1,752,384	3.26	6,465,887	3.25	0.250
median	3,795,072	5.51	9,134,570	5.50	0.501
3rd quartile	6,466,287	7.76	11,166,122	7.75	0.751
max	12,844,164	10.00	12,869,997	10.00	1.000

Table 3.3: Summary statistics for the M_w^s, PDI^s, M_w^l, PDI^l, and ϕ^l target attributes ($\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

	M_w^s [g/mol]	PDI^s	M_w^l [g/mol]	PDI^l	ϕ^l
count	200,000	200,000	200,000	200,000	200,000
unique count	189,398	200,000	197,912	200,000	200,000
mean	929,242	4.78	8,661,189	4.77	0.501
std	926,089	2.55	2,976,215	2.54	0.288
min	38,611	1.01	196,158	1.01	0.000
1st quartile	320,832	2.55	6,551,929	2.55	0.251
median	672,874	4.43	9,175,180	4.42	0.502
3rd quartile	1,197,026	6.82	11,174,497	6.80	0.751
max	11,539,282	10.00	12,869,958	10.00	1.000

This is reflected in the histograms and boxplots shown in Figure 3.9 for the no restrictions dataset and Figure 3.10 for the $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ dataset. Notably, M_w^s has a distinct exponential distribution in the restricted dataset, and the two M_w attributes no longer have a uniform distribution when added together, like they had in the no restrictions dataset. Interestingly the distribution of M_w^l is largely unchanged between the two datasets. The two PDI attributes also have an exponential distribution in the restricted dataset, with high PDI values appearing rarer than lower values. Contrastly, in the no restrictions dataset, both PDI attributes are uniformly

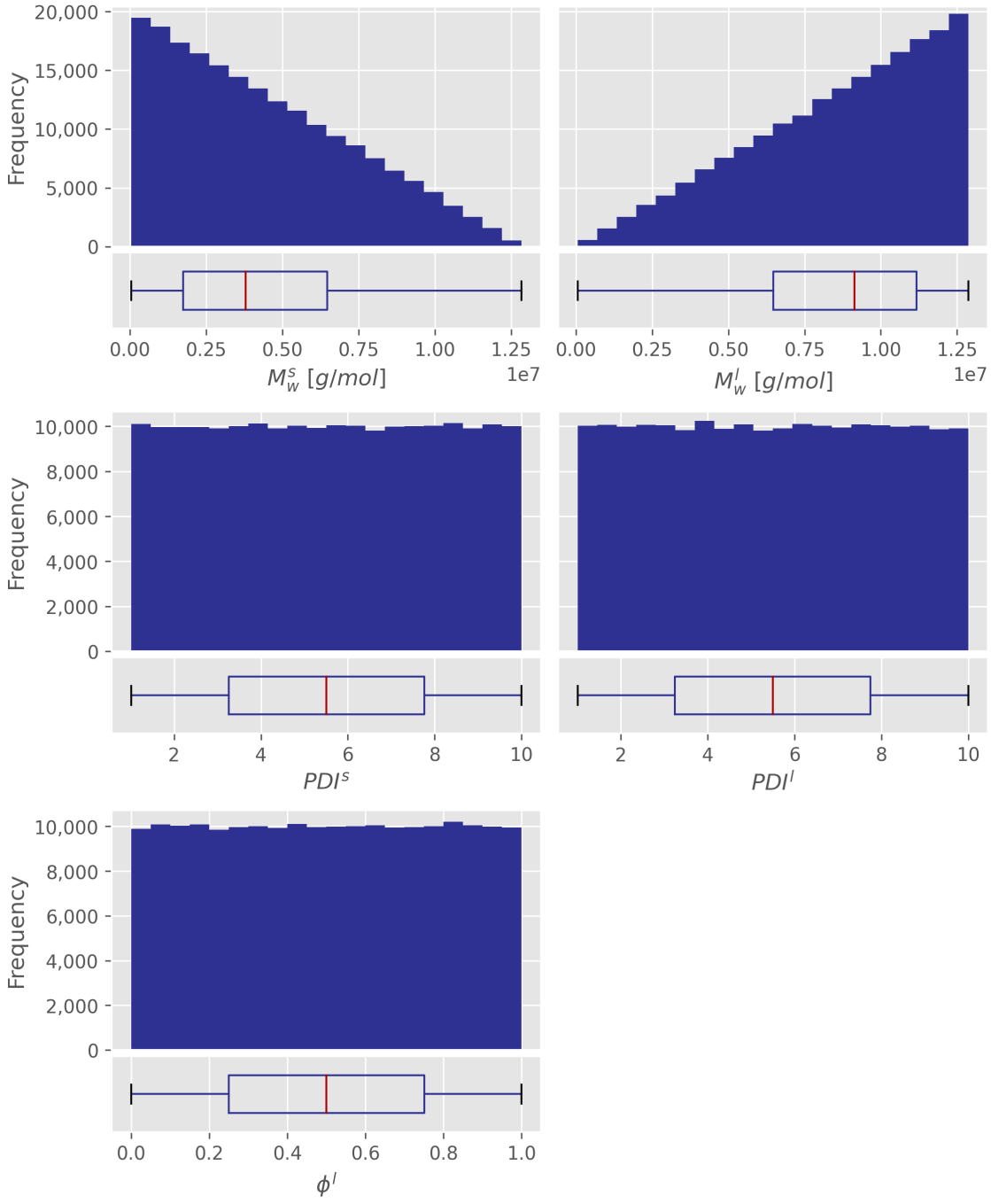

Figure 3.9: Histograms and boxplots of the M_w^s, PDI^s, M_w^l, PDI^l and ϕ^l target attributes (no restrictions bimodal dataset)

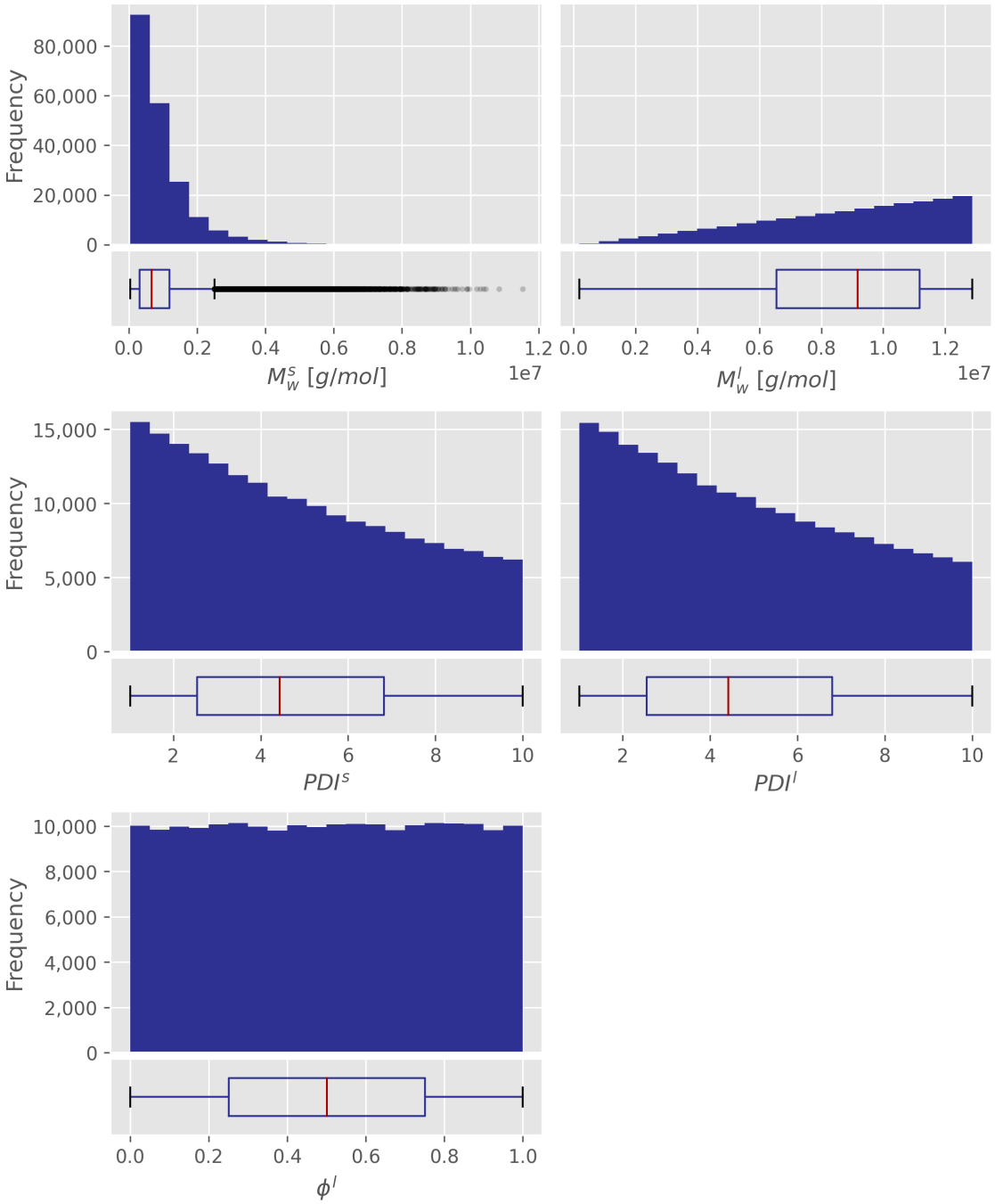

Figure 3.10: Histograms and boxplots of the M_w^s, PDI^s, M_w^l, PDI^l and ϕ^l target attributes ($\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

distributed. The summary statistics, histograms and boxplots for the two remaining datasets are not included here as they add little new insight not covered by the $M_w^s > PDI_{max}^1$ dataset. They can be found in the Appendix however.

When looking at the dataset's features, the differences were much less evident to the naked eye. Comparing the median and IQR curves of the no restrictions dataset (Figure 3.11) and the $M_w^s > PDI_{max}^1$ dataset (Figure 3.3) shows no immediately obvious differences between the datasets. The no restrictions dataset seems to have slightly lower values for G' and G'' in the mid and lower frequencies when examined closely, however. As in the unimodal dataset, we again see more spread in the low- and mid-frequency range data, while the variance in the high frequencies seems very small.

Due to the similarity of the features in all bimodal datasets, the rest of this section will only explore the $M_w^s > PDI_{max}^1$ dataset, as this was the dataset that was used the most in later stages of the project. All charts and tables shown were also created and checked for the other datasets, however, and can be found in the Appendix.

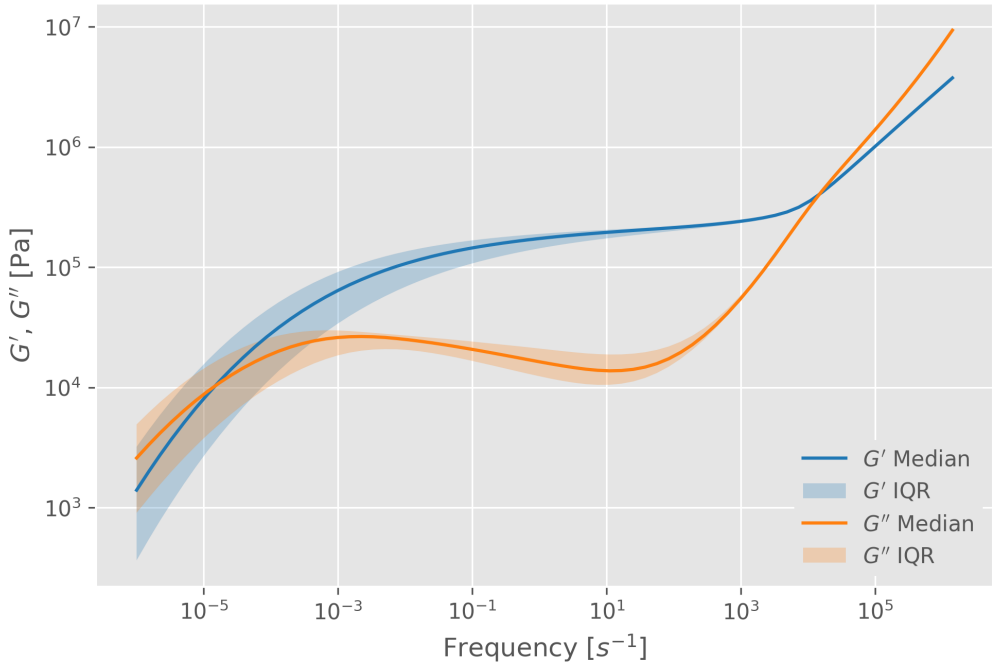

Figure 3.11: Median and IQR of the storage- and loss modulus curves (no restrictions bimodal dataset)

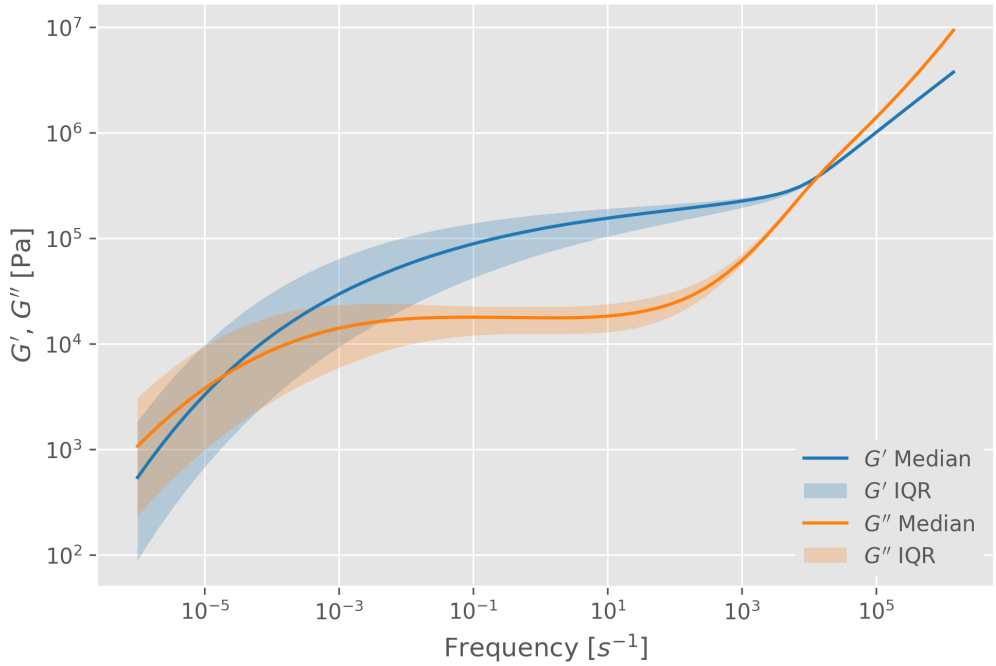

Figure 3.12: Median and IQR of the storage- and loss modulus curves ($\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

As in the unimodal case, the bimodal datasets contain 140 features, all of which contain float values. Table 3.4 shows the summary statistics of the feature variables. Similarly to the unimodal dataset, we can see that there are roughly 50% duplicates in the features. Furthermore, the much lower minimum values compared to the rest of the statistics make a return here as well, indicating there might be a substantial amount of breakout data points on the low end once more.

Table 3.4: Summary statistics for the G' and G'' features ($\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

	G' [Pa]	G'' [Pa]
count	14,000,000	14,000,000
unique count	8,282,335	7,451,884
mean	3.52×10^5	5.32×10^5
std	7.01×10^5	1.56×10^6
min	5.27×10^{-11}	1.93×10^{-4}
1st quartile	2.14×10^4	1.03×10^4
median	1.25×10^5	2.18×10^4
3rd quartile	2.32×10^5	7.84×10^4
max	3.75×10^6	9.55×10^6

To investigate this, we once more analysed boxplots of the lowest frequency, which can be seen in Figure 3.13. The chart also includes the same higher-end frequency we used for comparison in the unimodal case. Overall, this chart looks similar to the unimodal dataset, especially in the

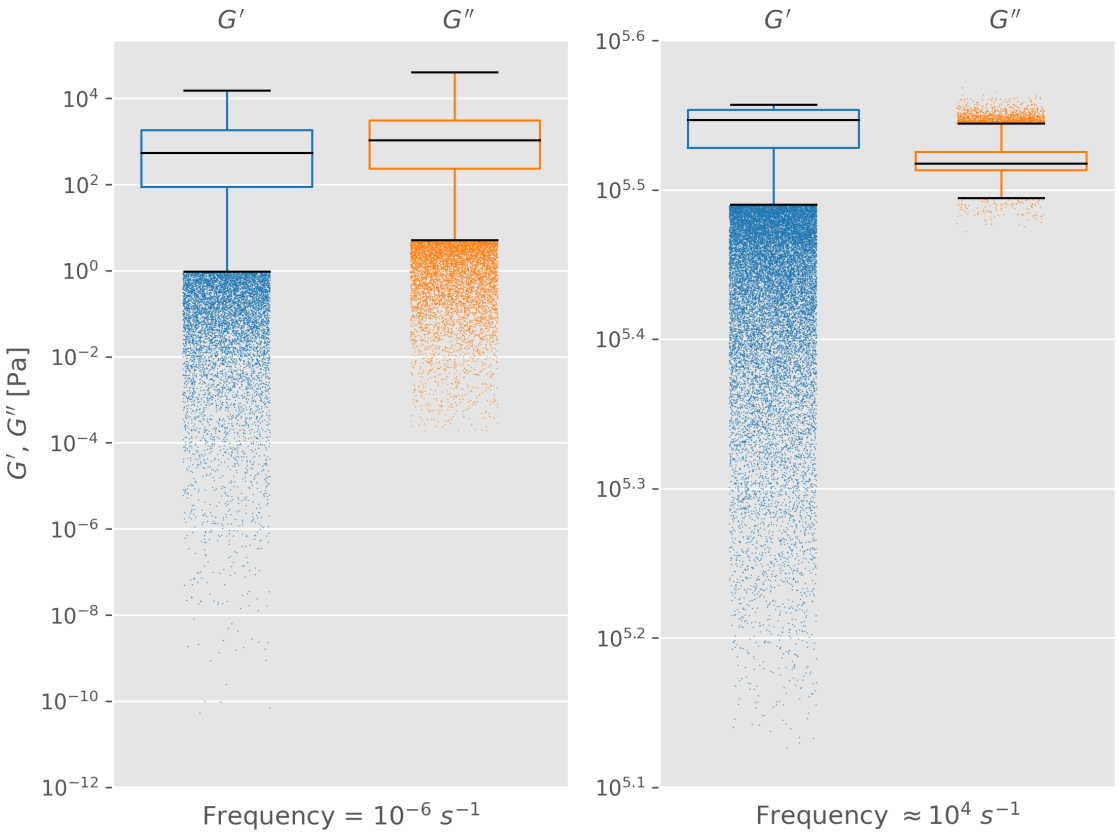

Figure 3.13: Boxplots showing the distribution of G' and G'' at the lowest frequency and at a high frequency ($\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

higher frequency where we have already seen minimal variation between instances and datasets. The distributions have a slightly larger IQR in this bimodal dataset, however. In the lowest frequency, the spread of the data looks similarly wide to what we saw with the unimodal MWDs, but there seem to be fewer data points outside the boxplot's whisker range. There are few data points where G' drops below 10^{-8} Pa this time around, and the distribution of G' and G'' are far more comparable than they were in Figure 3.4.

The last thing we looked at for the data exploration stage was the correlation among the features. As in Section 3.1.2 we examined a correlation matrix of all the features for this purpose, which can be seen in Figure 3.14. This chart looks nearly identical to the one in the unimodal case, so all observations made there also hold here.

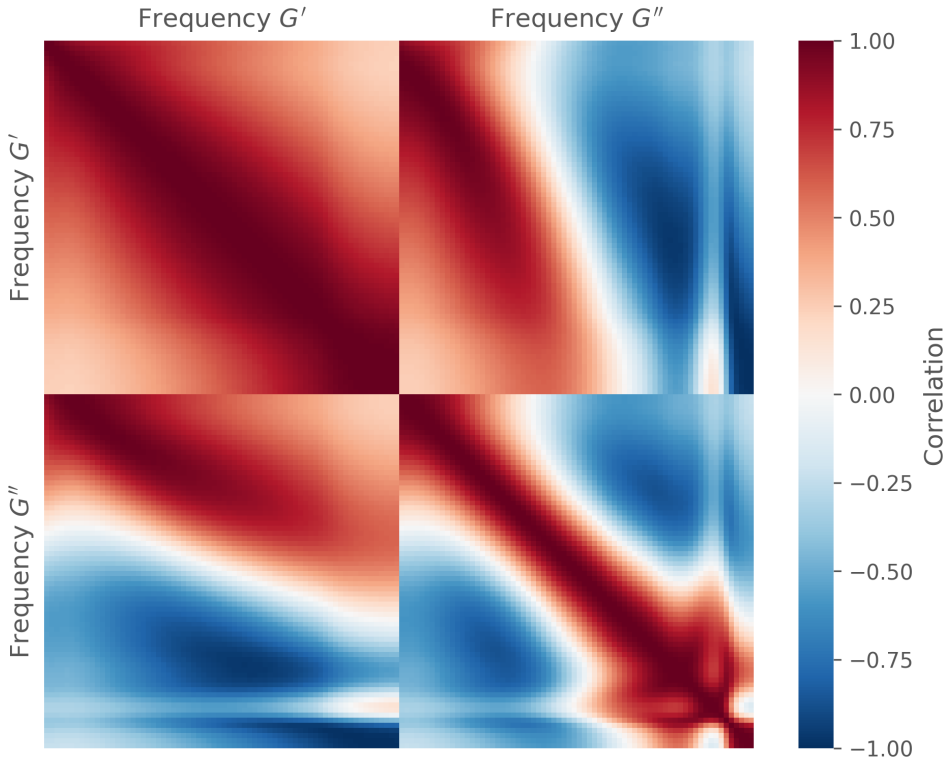

Figure 3.14: Correlation matrix of all features ($\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

3.2.3 Verifying Data Quality

As we saw in the unimodal case, the utilisation of software for the data collection led to exceptionally clean data. We did not see any sign of missing values, irregularities or unexpected values during our data exploration. We were, therefore, able to proceed to the modelling phase of the project using the four collected bimodal datasets without additional cleaning or data processing. Like with the unimodal dataset, we did scale the features of these datasets but did so as part of the modelling pipeline, which is covered in Section 4.2.

Chapter 4

Modelling

After exploring and preparing the data, the next step of this project was to build the machine learning models and experiment with the settings to find the optimal model architecture for our regression problems. This chapter first covers the initial model selection and then explains the modelling pipeline used throughout this project to feed the data to the models. Next, the final architecture for the unimodal MWD predictions is explained, and the optimisation process for arriving at the architecture as well as tuning the hyperparameters is discussed. The same discussion and showcase of the model architecture are then also provided for the bimodal regression problem. Finally, the performance metrics used to evaluate the models are presented.

4.1 Model Selection

Both the unimodal and bimodal prediction tasks are multi-target problems, as the goal is to predict either 2 (unimodal) or 5 (bimodal) targets simultaneously. Though all regression models could be used as predictors for these problems by fitting a new model for each target attribute, we decided to use models that support multi-output predictions. This way, only one model had to be trained at a time, which was a much cleaner solution, especially in the bimodal case. Often the outputs of a multi-output prediction depend not only on the inputs but also on the other outputs and are thus not independent from each other. Unlike the ”training separate models for each target” approach, multi-output models can account for this. Several regression models natively support multi-output predictions, including decision trees, random forests and neural networks, when appropriately constructed. We performed some baseline testing to determine which model we wanted to finetune for this project. Table 4.1 shows performance metrics for a decision tree model, a random forest model, and a multilayer perceptron neural network. The metrics we used to evaluate the models throughout this project are explained in Section 4.5, but

Table 4.1: Mean absolute error (MAE), mean relative error (MRE) and the averaged MRE across all targets (Avg. MRE) for decision tree, random forest and neural network model predictions (unimodal dataset)

	MAE (M_w)	MAE (PDI)	MRE (M_w)	MRE (PDI)	Avg. MRE
Decision Tree	36,896	0.050	0.91%	3.03%	1.97%
Random Forest	13,635	0.025	0.33%	1.86%	1.09%
Neural Network	12,621	0.024	0.42%	1.14%	0.78%

for now, the critical thing to note is that all the metrics shown are error metrics, where lower values are better. As we can see from the results above, the random forest model and the neural network have comparable performance, while the decision tree performs substantially worse in all metrics. The neural network did have the edge in 4 out of 5 performance metrics in the tests we ran. However, the gap is small enough that either model could feasibly perform better after hyperparameter tuning. We chose to use neural networks for the rest of this project as we thought it might give us more freedom with tuning due to the substantial possibilities of different architectures.

4.2 Machine Learning Pipeline

As we have seen during the data exploration (Section 3.1.2), both the targets and features have different ranges of values. It was, therefore, essential to prepare the instances correctly before feeding them to the machine learning models. Neural networks generally require features to be scaled for optimal performance as, otherwise, one feature could gain too much importance in setting the weights over the rest due to containing larger values. The high-frequency features would, therefore, have more influence than the low-frequency features, which would be especially bad in this project due to the low variance in the high-frequency regions. We, therefore, made sure to apply min-max scaling to the features before passing them to the neural networks. Min-max scaling scales all values of a given feature to range between 0 and 1, with the original max value becoming 1 and the original min value becoming 0.

The targets also had to be scaled, as M_w and PDI values have very different value ranges. Neural networks are trained by tuning weights to try and minimise the loss function. The loss in regression problems is typically calculated from the error or difference of each prediction from the true value. An error of 5 for a M_w prediction would be superb, while it would be rather bad for a PDI prediction where values only range from 1.01 to 10. To ensure all targets are weighted the same for the loss calculation, we also wanted to ensure the targets are scaled using min-max scaling.

One more thing to pay attention to was splitting the data into training and validation sets and at which point to apply the scaling. We typically split our datasets to use 10% or 20% of the instances for validation and testing, depending on the sample size. It is vital that any transformation of the validation set is only dependant on the training data, not the validation data itself, to ensure reliable results. To perform this cleanly, we set up a pipeline that would handle the preprocessing for all models at the training and prediction stage and thus made repeated testing much more streamlined. For example, when training a model, the pipeline took the unedited training features and targets as inputs. It then scaled these features and targets using the min and max values of the provided training data. Data is typically fed to neural networks in batches, so the next step was to separate the data into batches that could then be fed to the model for training. Feeding all batches to the neural network is known as one epoch. We repeated this process for as ever many epochs we wanted to train for. After training, the pipeline also took care of preparing the validation data to be fed to the model to make predictions. Crucially, the validation data was scaled using the min and max values discovered from the training data to ensure the predictions were only influenced by the training data. When making predictions in practice, we might only have one instance of features, but the data would still need to be scaled for the model to work as designed. As a last step when making predictions, the results were scaled back up to the original scale to allow for more readable results and easier comparison with the validation targets.

4.3 Unimodal Model Architecture and Hyperparameter Tuning

Neural networks have infinite configuration options due to their modular architecture. This section first covers the layer architecture we ended up with for the unimodal regression problem before explaining how we arrived at that design and chose other hyperparameters. Figure 4.1 shows the final design of the neural network. The chosen design consists of 8 layers. The first layer is the input layer consisting of 140 neurons, one for every feature of an instance. This

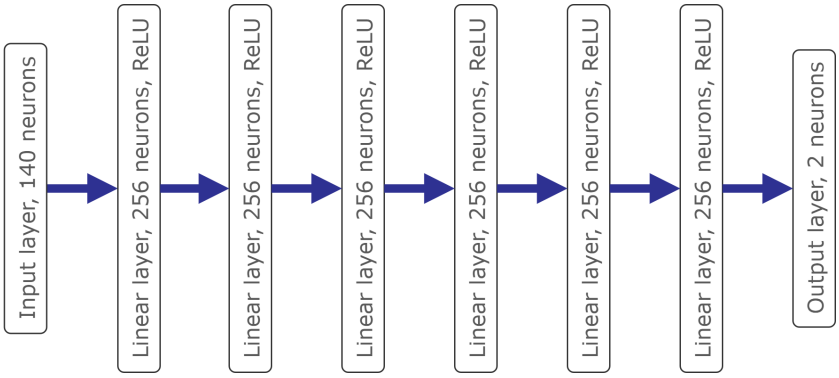

Figure 4.1: Neural network architecture (unimodal MWD)

layer is followed by 6 hidden layers, each with 256 neurons. Every hidden layer contains an activation component where the output of a neuron is determined by the weighted sum of the inputs and biases. This activation function is responsible for introducing non-linearity to the output. We used rectified linear units (ReLU) as the activation function for all the hidden layers in our neural network. The final layer is the output layer that contains two neurons, one for each target variable.

The process of determining the model architecture and setting the hyperparameters was iterative, as we did not have access to the computing power required to exhaustively grid-search optimal parameters. This means there could be a more optimal set of parameters, as we did not check every possible combination of settings. We are, however, happy with the optimisation achieved and were able to improve performance by tweaking several parameters. This included architectural decisions like adding more layers to the network or trying different numbers of neurons per layer. We also tried adding convolutional layers to help improve performance. Here there were two options, either use 1D convolutional layers on the 140 features or stack the G' and G'' features to form 2D tensors with dimensions (70, 2) and use 2D convolutional layers. In the latter case, G' and G'' influence each other by the filter passing over both values at once. We did not encounter any problems with overfitting in the unimodal case, probably because using 150,000 instances was more than enough data to prevent this. Consequently, we did not add any regularisation components to the network, such as dropout or normalisation layers.

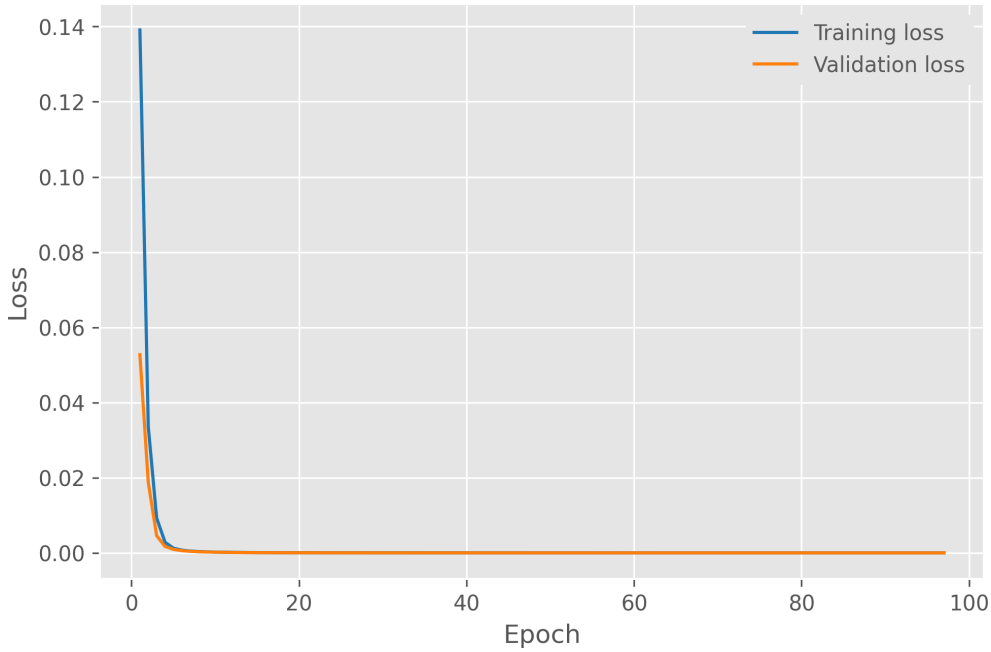

Figure 4.2: Training and validation loss over training epochs (40,000 training instances, unimodal dataset)

Besides the architectural choices, there were several hyperparameters to choose. One of which was the **number of epochs** we wanted to train the model for. The optimal number of epochs can change drastically depending on the architecture and other hyperparameters. We, therefore, chose a dynamic approach to using this parameter during the optimisation process. The goal in choosing the number of epochs to train for is to train long enough to reduce the loss as much as possible without overfitting. We can monitor this by looking at the train and validation loss during training, as is visualised in Figure 4.2.

Generally, the model is still improving as long as the training loss keeps decreasing. The validation loss tracks the loss of unseen instances the model was not trained on. If the training loss keeps decreasing, but the validation loss starts to increase, this is a sign of overfitting. In this case, the model is starting to fit to the inherent noise in the training data and does not generalise as well to unseen data anymore. If we keep training for too long, the model will get better and better at predicting the training instances but worse at predicting the unseen validation instances. We generally want to stop training when the validation loss stops decreasing, therefore. As this point can vary when we change other parameters, we used early stopping to track the validation loss during training and automatically stopped training when it did not decrease in 5 epochs. It may seem from Figure 4.2 that both the training and validation loss do not decrease after epoch 10, but this is due to the loss decreasing so drastically in the first few epochs. The chart looks different if we look at the same data but do not include the first 35 epochs. This can be seen in Figure 4.3 where both loss curves keep decreasing until flattening out at almost 100 epochs.

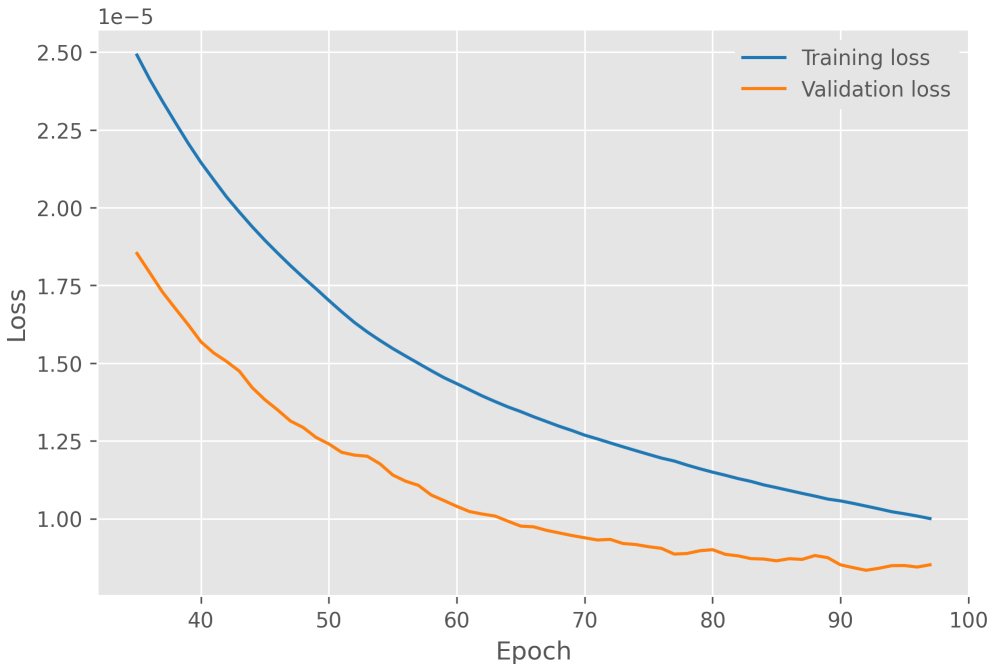

Figure 4.3: Training and validation loss over training epochs, showing epoch 36 and above (40,000 training instances, unimodal dataset)

A hyperparameter that is tied closely to epochs is the **batch size**. As we have noted before, the data is typically fed to the neural network in batches. The trainable parameters of the neural network get updated based on each batch that is passed to the network. Larger batches, therefore, lead to fewer updates of the parameters per epoch. Larger batches can be more efficient to minimise the loss function, however, even though this might require more epochs, as each epoch will also take less time to finish. We tried a few options of batch sizes (64, 128, 256, 512) before settling on a batch size of **128**.

We have talked a lot about tweaking the trainable parameters of the neural network to minimise the loss function. This optimisation is performed by an **optimiser** that tries to reduce the loss by taking iterative steps towards an optimal solution. At each iteration through the network, the optimiser calculates gradients for all the trainable parameters to determine how to tweak them to reduce the loss effectively. There are many options of optimisers to use with neural networks, of which we tried stochastic gradient descent (SGD), Adagrad and AdamW. We settled on using the **AdamW** optimiser in our neural network after testing these options.

Tied to the optimiser is the **learning rate**. This hyperparameter controls how big the steps towards the optimal solution taken by the optimiser should be. This is a critical hyperparameter to get right, as taking too large steps can result in overshooting the minimum of the loss function and never reaching the global minimum. Too small steps, on the other hand, can take too long to train and potentially result in getting stuck in a local minimum when searching for the global minimum of the loss function. Tuning this hyperparameter resulted in the biggest performance gain during our optimisation process. We tried learning rates of 10^{-3}, 10^{-4}, 10^{-5} and 10^{-6}, ultimately settling on $\mathbf{10^{-5}}$.

Finally, we also had a choice of which **loss function** to use in order to calculate the loss. The options here depend on the type of prediction problem we are dealing with. For regression problems, the two main options are using mean absolute error (MAE), often referred to as L1 loss, or mean squared error (MSE), often called L2 loss. The formulas for calculating MAE and MSE are covered in Section 4.5. Usually, using MSE is preferred unless there are many outliers. As we did robust data exploration without finding clear outliers, we were confident that MSE was the right choice of loss function for our model. However, we tried MAE quickly as a sanity check and confirmed that performance was better using **MSE**.

4.4 Bimodal Model Architecture and Hyperparameter Tuning

As we did in the unimodal case, we first cover the neural network architecture we ended up using for the bimodal MWD predictions in this project. We then follow that up with the various hyperparameters we tested, as well as the selection we decided on using.

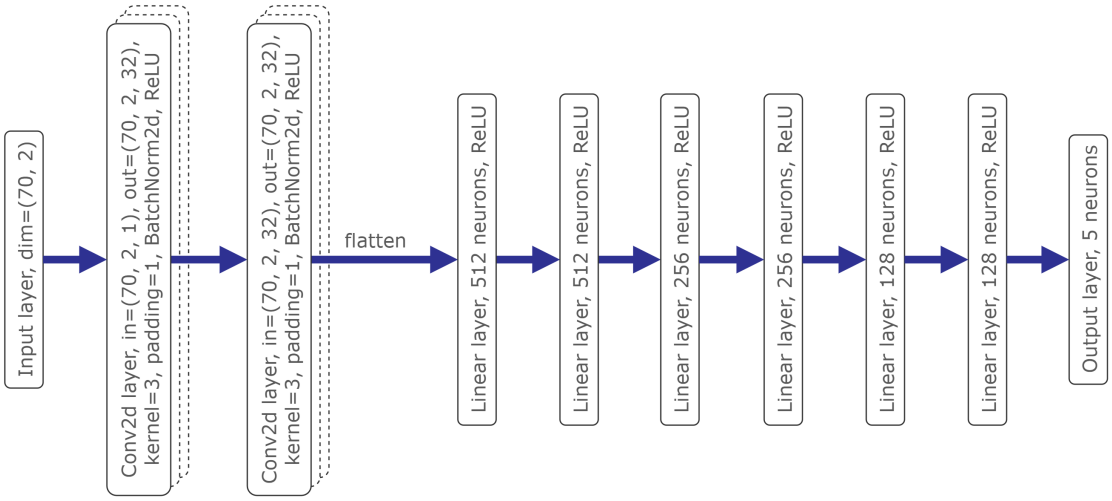

Figure 4.4: Neural network architecture (bimodal MWD)

Figure 4.4 shows the design of the neural network we settled on for the bimodal case. As we can see, this neural network has a slightly more complicated design, including convolutional layers, which we did not end up using in the unimodal case. This time we did not input the 140 features as one array of values but instead stacked G' and G'' to form a two-dimensional input of size 70x2. This is followed up by two convolutional layers. In these layers, a convolution mask, known as kernel, of size 3x3 is moved over the two-dimensional input, recalculating each point based on the mask and the surrounding values. We used padding to add zeros wherever the mask went over the edge of the input. The output after passing the convolution mask over the input is known as a feature map. Typically we pass multiple different convolutional masks over the input to extract different features from the data, resulting in multiple feature maps. Both convolutional layers used in our design output 32 feature maps by using 32 separate convolution masks. These masks are initialised randomly and are part of the trainable parameters of the network, meaning they get tuned alongside the other parameters of the network. The feature maps are then batch normalised before being passed through an activation function. This was done to add regularisation to the network due to overfitting occurring when we were testing various architectures. The activation function we used to introduce non-linear transformation to the feature maps was once more the ReLU. The output of the convolutional layers is flattened to a one-dimensional array to be used with the linear layers that follow. Similarly to the design for the unimodal MWDs, there are 6 linear layers that include ReLU activation functions. The layers have varying numbers of neurons this time around, though, getting smaller towards the output layer. The output layer contains five neurons for the five target variables we are trying to predict.

The modelling process of trying different architectures was once again an iterative process, where we tried many possible designs. Adding layers to increase the complexity of the model, adding batch normalisation when we encountered overfitting and so on. We also tried non-

convolutional architectures and 1D convolutional layers as well before arriving at this design.

The same remaining hyperparameters we saw in Section 4.3 also needed to be chosen for the bimodal model. The options for each parameter were mostly the same, so we will not explain the process and the reasoning again. Instead, the list below shows the parameters we tried as well as the chosen values in bold.

- **Batch size:** 64, **128**, 256, 512
- **Optimiser:** SGD, Adagrad and **AdamW**
- **Learning Rate:** 10^{-3}, 10^{-4}, $\mathbf{2 \times 10^{-5}}$, 10^{-5}, 10^{-6}
- **Loss function:** MAE and **MSE**

As we can see, the only hyperparameter that was chosen differently from the unimodal case was the learning rate. This change was done to speed up training a bit as the results were very similar to using a learning rate of 10^{-5}. Finally, we again used early stopping to determine the optimal number of epochs, which can be seen in Figure 4.5. This time we waited for ten epochs of non-increasing validation loss, however, since we often saw training stopping too early when using a threshold of five epochs.

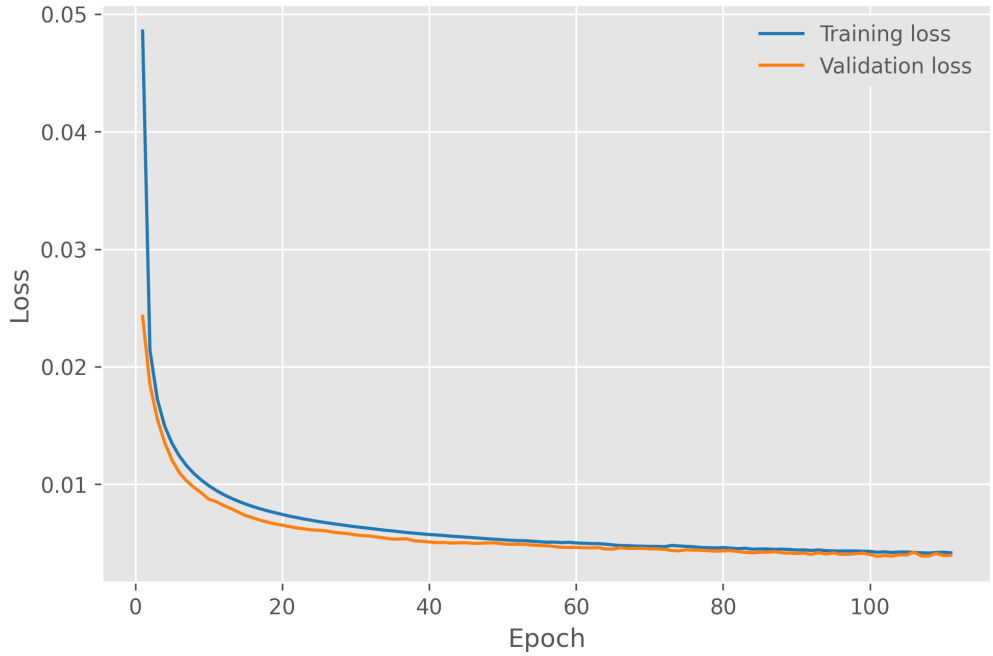

Figure 4.5: Training and validation loss over training epochs ($\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

4.5 Performance Metrics

In order to optimise the models and evaluate their performance, we needed to define appropriate performance metrics. These metrics were used consistently throughout this project to ensure that performance numbers were always comparable between tests and models. We could have used the mean squared error (MSE), which we used for the loss function of both the unimodal and bimodal neural networks. The formula for calculating MSE is shown below, where n is the number of instances, y is the true value and $\hat{y}$ is the prediction for a given instance (Chugh, 2020).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2$$

However, MSE is not a very intuitive error metric to interpret as the errors of predictions are squared during its calculation. Instead we preferred using the **mean absolute error** (MAE) for evaluating the models, the formula of which can be seen below (Chugh, 2020).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}|$$

Unlike MSE, MAE retains the original units of the values, making the errors much more intuitive and readable. In the case of M_w and PDI values, using MAE is not always the best error metric, however. An absolute error of 30,000 g/mol might be quite good when the true value is in the high molecular weight range, e.g. M_w = 12,000,000 g/mol. However when the true value is small, e.g. M_w = 40,000 g/mol, being off by 30,000 g/mol is much more significant. To account for this, we also used the **mean relative error** (MRE) to judge the performance of our models. The formula to calculate the MRE is shown below (Helmenstine, 2018).

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}|}{y_i}$$

When we are dealing with PDI values, the formula needs to be modified slightly. As PDI values start at 1 instead of 0, we subtracted 1 from the denominator when calculating the MRE of PDI values, as shown below.

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}|}{y_i - 1}$$

We generally used MAE to get a feeling for how significant the errors were. For judging model performance and making optimisation decisions, we used MRE, however, as it is the more appropriate metric given the problem of predicting MWDs. As all of these metrics are error metrics, lower values indicate better performance. An overview of the performance of the models is provided in Section 5.1, where the differences of using MAE or MRE are also outlined.

Chapter 5

Evaluation

This chapter provides an analysis of the prediction performance of the neural networks we constructed in Chapter 4. First, the performance for each of the target variables is investigated. We also look into which regions of a target's range are harder to predict. Doing this, we gain an understanding of the model's accuracy in various regions of the target attribute space. Next, we analyse the prediction performance using variously sized samples of the dataset to see how the performance changes with larger datasets. We then investigate how the prediction accuracy changes when we restrict the frequency range used for the G' and G'' features. Finally, we also examine the effect of varying the proportion of long-chain polymers of the compounds in the bimodal case.

5.1 Model Performance

A unimodal MWD can effectively be described using two parameters. The M_w gives information on the location of the distribution, while the PDI provides information on the spread of the distribution. We, therefore, analysed the prediction performance of these target variables thoroughly to determine whether a machine learning approach using flow data is a feasible way to determine the MWD of a plastic.

We looked at the absolute errors of the predictions first to get a feel for how far off the predictions were from the true values on average. Figure 5.1 shows the absolute errors for all the predictions made on the testing set of the unimodal dataset. The model in question was trained on 40,000 training instances. Every data point is positioned on the x-axis by its corresponding true value. This way, we could analyse whether there were regions of the target range where the model performed better or worse. Before going in-depth on the charts, however, we can look at the MAE values to see how well the model performed on average. We can see that the MAE for

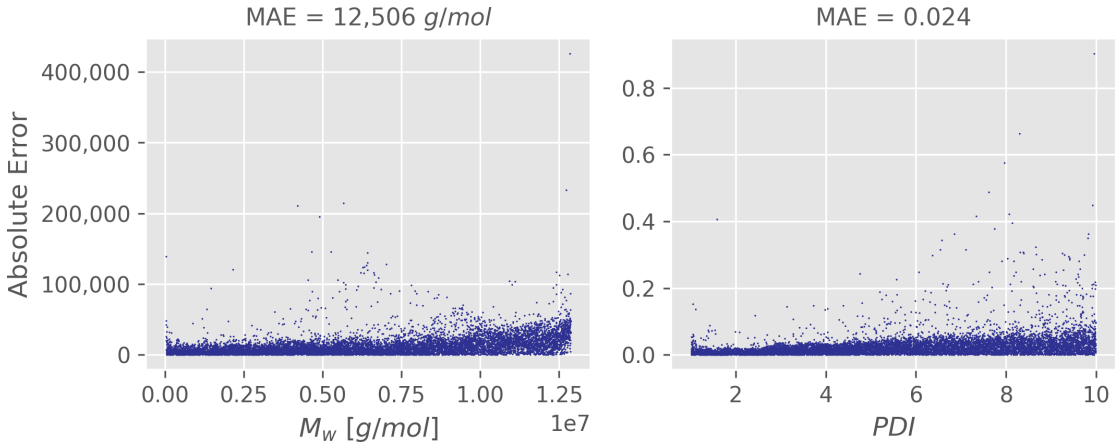

Figure 5.1: Absolute errors of M_w and PDI across their true value range (40,000 training and 10,000 testing instances, unimodal dataset)

predicting M_w was 12,506 g/mol, which is less than the M_e of polystyrene. Combined with the MAE for predicting PDI of 0.024, these are very promising results. Looking at the distribution of absolute errors across the target range, we can see that the errors tend to get larger towards the right side of both charts. This indicates that high M_w and PDI values are slightly harder to predict accurately. However, this effect does not seem nearly strong enough to be a problem when thinking of the relative errors of the two targets. In fact, it is in the low ranges for both the M_w and the PDI where we can expect to see the highest relative errors. The absolute errors are only marginally smaller on the left side than the rest of the charts for both targets, and they even seem to increase on the far left side. Of course, we can create the same charts for the relative errors in order to visualise this, which can be seen in Figure 5.2.

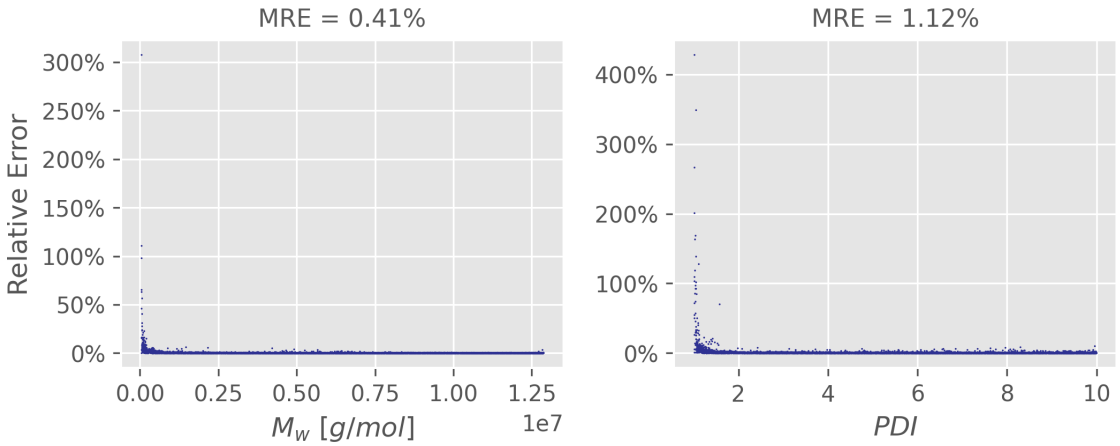

Figure 5.2: Relative errors of M_w and PDI across their true value range (40,000 training and 10,000 testing instances, unimodal dataset)

As expected, the relative errors peak in the lowest area of the target range. These charts may be a bit misleading, however, since just a few data points distort the entire chart. If we ignore the far left of the charts, the relative error is actually impressively low across the entire range for

both the M_w and PDI attributes. This is also reflected in the MRE metrics, with the MRE for M_w coming in at 0.41% while the model achieved an MRE of 1.12% for predicting the PDI. This results in an average MRE across all target attributes of 0.76%, which is a very respectable result.

From the results so far, we could gather that the prediction accuracy is best in the middle of the target attribute space, only declining somewhat on the edges of the space where either M_w or PDI is very small or very large. We checked this through the visualisation in Figure 5.3 where we used an even grid of values across the target attribute ranges as the testing instances and plotted the model predictions on top of them.

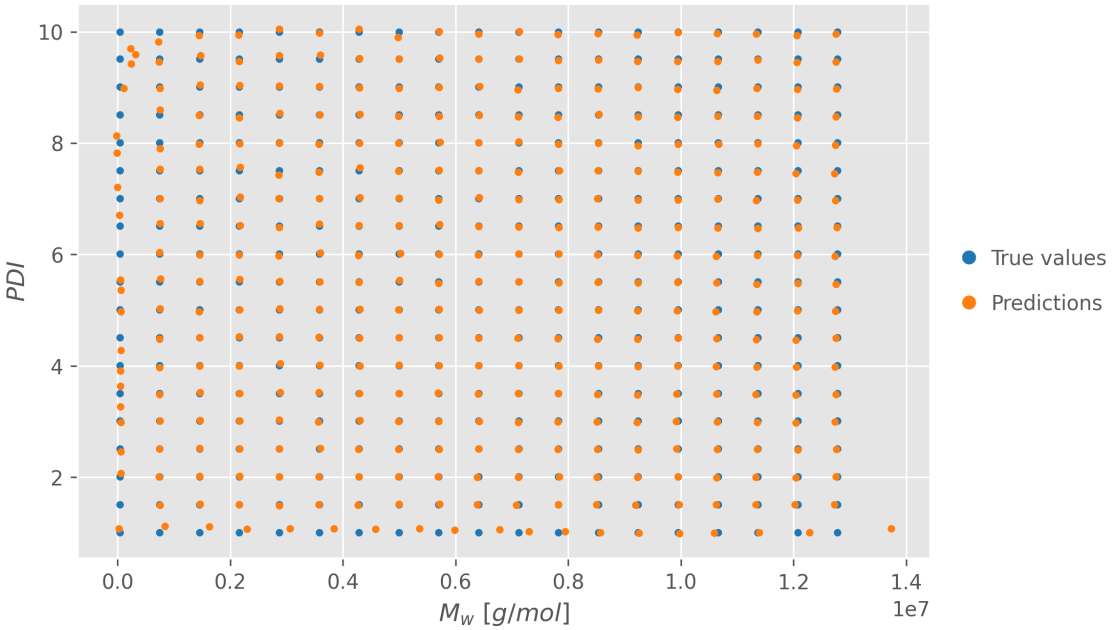

Figure 5.3: Predictions vs. true values showing the accuracy in various regions of the target attribute space (40,000 training and 361 testing instances, unimodal dataset)

Interestingly this chart indicates that it is really only the lowest parts of the two target ranges where the largest errors happen. In the area where the true values for both M_w and PDI are large (top right of the chart), we only see minor errors being made by the model. This most likely means the high error data points on the right side of the charts we saw in Figure 5.1 were associated with a low value of the other target variable. A good example of this is the last prediction on the bottom right in Figure 5.3. The prediction for the M_w is off by a wide margin. However, this is also a prediction in the lowest PDI region, where we see larger errors being made across the entire M_w range. In conclusion, prediction accuracy seems to suffer whenever one of the two target attributes is very small. In the remaining target attribute space, we see very strong prediction performance, which resulted in the impressive MRE metrics for the unimodal model.

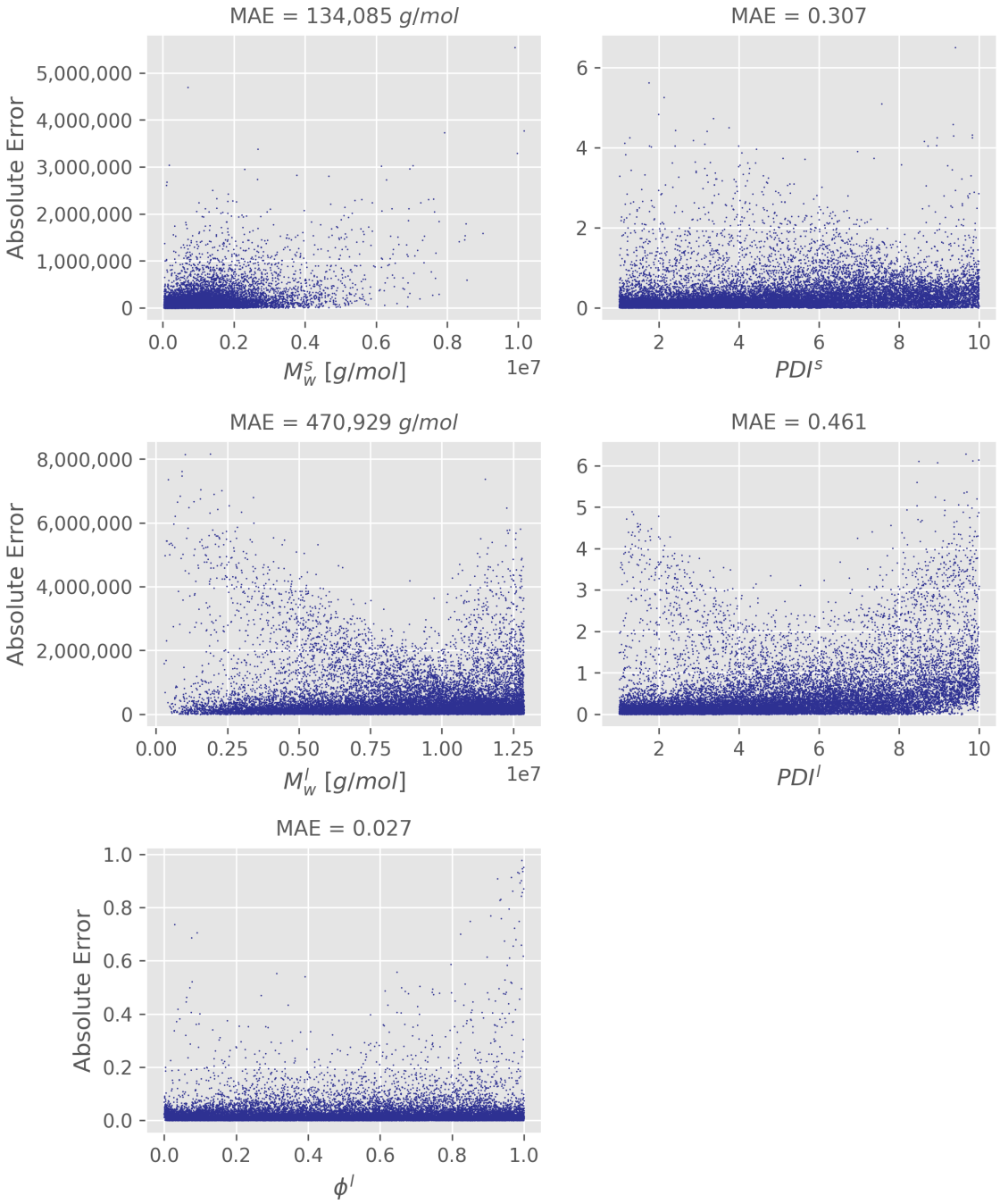

Figure 5.4: Absolute errors of M_w^s, PDI^s, M_w^l, PDI^l and ϕ^l across their true value range (180,000 training and 20,000 testing instances, $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

Shifting our focus over to the bimodal prediction problem, recall that here we are dealing with five parameters to predict, as the MWD has two peaks. The five parameters are the M_w and PDI for the short- and long-chain peaks, as well as the proportion of long-chain polymers in the compound.

Figure 5.4 shows the absolute errors for all the predictions made on the testing set of the $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset. The same charts for the other bimodal datasets can be found in the Appendix. The neural network used for these results was trained on 180,000 training instances. The first takeaway here was the substantially larger errors across the board compared to the unimodal case. The MAE for both M_w parameters is well over 100,000 g/mol, while the MAE for both PDI values is over 0.3. The MAE for ϕ^l also looks comparable to that of PDI^s, given its different scale from 0 to 1. Looking at the distribution of the absolute errors across the target ranges, things get a little more complicated than in the unimodal case. As we saw in Section 3.2.2, the true values are no longer uniformly distributed across the possible ranges in this dataset. For instance, the values for M_w^s tend to occur much more frequently in the lower part of the possible range of M_w values, which is also reflected in Figure 5.4. Here we can see some of the largest errors towards the left side of the chart, which indicates we are once more dealing with worse performance in the lowest regions of the target range. This is especially unfortunate in the case of M_w^s since so many of its true values are concentrated there. The absolute error charts for PDI^s, M_w^l and PDI^l are quite similar, with absolute errors increasing slightly in the higher ranges. We also see higher errors towards both the left and right sides of the charts, indicating the best performance is once again achieved in the middle parts of the target ranges. Looking at the absolute errors for ϕ^l, they seem mostly uniformly distributed across the target range. There do seem to be some larger errors towards the higher end of the range, however.

Figure 5.5 shows the relative errors for the M_w^s, PDI^s, M_w^l and PDI^l predictions. ϕ^l was not included here as using relative error for this attribute does not really make sense. An absolute error of 0.2 for ϕ^l is equally bad if the true value is 0.1 or 0.9. Looking at the MRE for the four target attributes, we can see the higher absolute errors are also reflected here, with all but the MRE for M_w^l coming in at over 20%. The average MRE across the four targets was 20.92%, which is much higher than the 0.76% achieved in the unimodal problem. While not surprising, this does show how much more difficult predicting bimodal MWDs is. There is a silver lining, though, in the prediction performance of M_w^l, which is much better than the rest of the target attributes. As the long-chain peak is likely more important for dictating a materials physical properties, and this being the attribute that defines the location of that peak, the higher performance is encouraging. Additionally, the PDI values we would likely encounter in practice for bimodal MWDs probably range more between 2 and 4 rather than the extreme values we also allowed here for consistency. In this range, the prediction performance for both PDI^s and PDI^l look more promising. The distribution of relative errors looks similar to what we saw with the

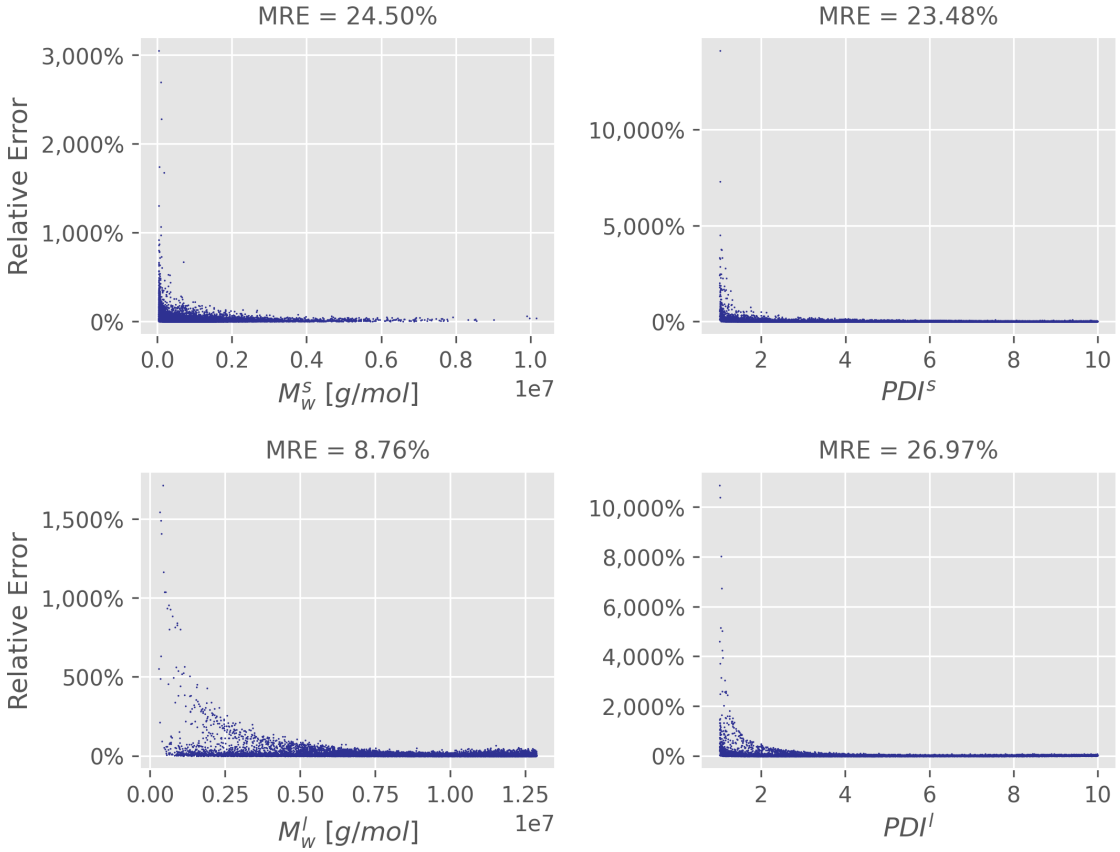

Figure 5.5: Relative errors of M_w^s, PDI^s, M_w^l and PDI^l across their true value range (180,000 training and 20,000 testing instances, $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

unimodal MWDs, with the relative errors for all four target attributes peaking on the left side of the charts. Again, this can be somewhat misleading since a relatively small number of data points distort the charts with very high errors in all cases. This also affects the MRE values as they use the mean for their calculation instead of the median that is more robust to outliers.

In conclusion, the prediction performance in the bimodal case is substantially worse than in the unimodal case. We again see the best performance in the middle of the target ranges, with the highest errors occurring in the extreme values. In terms of relative errors, it is once more the lowest true values that are related to the highest errors, though this is not uncommon with relative errors due to their formula.

5.2 Training Set Size

One aspect we wanted to investigate further was varying the training set size. We were interested in both understanding how the prediction performance changes with differently sized training

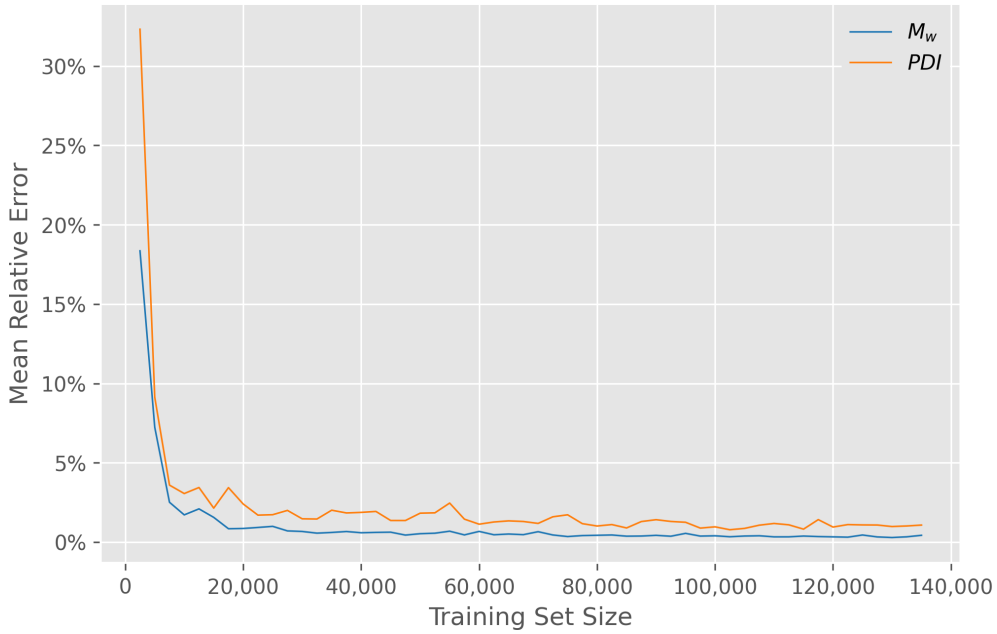

Figure 5.6: Mean relative error for the target attributes M_w and PDI across various sizes of training sets (15,000 testing instances, unimodal dataset)

sets and finding the point of diminishing returns where further increasing the training set size is no longer worth the effort of collecting the required additional data. Figure 5.6 shows the MRE of both target attributes in the unimodal case at various training set sizes. As we can see, the MRE for both targets initially decreases drastically with increasing training set sizes. After reaching a training set size of around 25,000 instances, the improvements become much slower, however. From this chart we can conclude that a training set size of at least around 25,000 instances is recommended for the neural network we used to predict the unimodal MWDs. This is also why we only used 40,000 training instances of the unimodal dataset in many of the tests and examples throughout this project, as using more resulted in negligible performance increases while increasing training time significantly. The point of diminishing returns depends largely on the cost of collecting data points. While generating the data did take considerable time and computing power using the software approach, the cost was minuscule compared to gathering the data from real-world oscillatory shear experiments. In that case, a training set size of 7,500 instances might be the more sensible target to aim for as that is where we see the strongest kink of the MRE curves in Figure 5.6.

If we look at the same chart for the $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset (Figure 5.7), we can see a similar picture, but this time the initial performance gains are not quite so drastic. The MRE for most targets seems to decrease at about the same rate, except for M_w^l, but this is due to its already much lower MRE at the smallest training set. Based on this chart, using at least 40,000 training instances is recommended to train the bimodal neural network we presented in Section 4.4. However, as the errors are still very high across the board in the bimodal case,

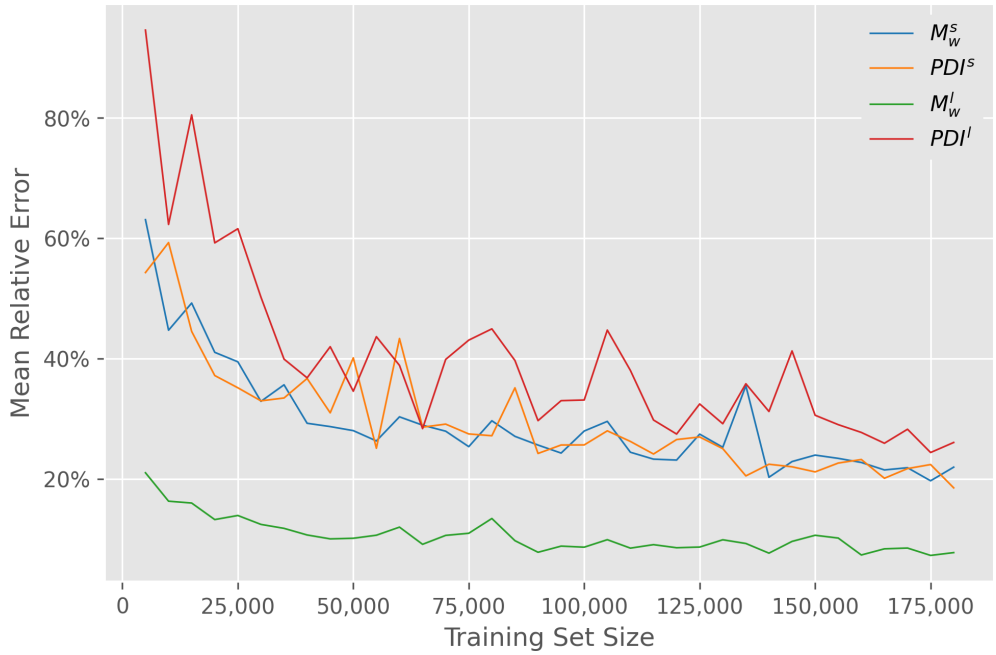

Figure 5.7: Mean relative error for the target attributes M_w^s, PDI^s, M_w^l and PDI^l across various sizes of training sets (20,000 testing instances, $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

it is worth using as much data as possible since the errors continue to decrease with larger training sets, just less quickly than on the left side of the chart. For this reason, we always used the complete datasets when training the bimodal models. The charts showing the MRE of the targets at varying training set sizes for the other bimodal datasets can be found in the Appendix.

5.3 Peak Separation of Bimodal MWDs

We stated in Section 3.2.1 that the reason we collected four datasets for the bimodal case was to investigate variously strict restrictions on how close the peaks of the MWD were allowed to be. We have mostly used the $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ dataset so far, however, and moved most charts for the other datasets to the Appendix. This was done partly because there was significant overlap between these charts, and including them all for every analysis we undertook would have provided little additional insight. But also for some of the more computationally advanced investigations, like analysing the prediction performance at various frequency ranges in the next section, we lacked the computational resources to perform these for all datasets. The reason for choosing the $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ dataset to focus on becomes apparent when we compare the prediction performance of the four bimodal datasets. Building on the insights from the previous section, Figure 5.8 shows the averaged MRE of the four target variables M_w^s, PDI^s, M_w^l and PDI^l for each of the bimodal datasets. We can see from this chart that the prediction performance improves at the same rate for all datasets with increasing sizes of training sets. Crucially though,

Figure 5.8: Averaged MRE of the M_w^s, PDI^s, M_w^l and PDI^l target attributes across various sizes of training sets for the four bimodal datasets (20,000 testing instances)

the average MRE for the "no restrictions" dataset is roughly 20% higher than the other datasets across the board. The three restricted datasets all show very similar prediction performance, however. As the $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ dataset sees vastly improved prediction performance while being the least restrictive dataset of the three modified options, we chose this dataset as the main focus for the bimodal modelling. Figure 5.8 also clearly shows that the models struggle when the two peaks are allowed to be too close together. As we already discussed in Section 3.2.1, when the peaks are very close together, the MWD becomes almost indistinguishable from a unimodal distribution to the naked eyes, so the models struggling with that situation is not surprising.

5.4 Frequency Ranges

We saw in Section 3.1.2 that the datasets contain a total of 70 features each of G' and G'' values at frequencies ranging from $10^{-6}s^{-1}$ to $10^{6}s^{-1}$. Furthermore, we saw that the spread of the data varies at different frequencies, with the high-frequency region being especially interesting due to a lack of variance in the G' and G'' values there. We, therefore, wanted to investigate if the prediction performance varies when we restrict the frequency range of the features to various regions. We had 70 features, so we decided to cut the complete frequency range into 7 pieces with 10 features each. We then tested the model's prediction performance at every possible consecutive combination of these ranges. This resulted in 28 different frequency ranges, for which we ran the performance tests. Figure 5.9 summarises the findings in a handy visualisation.

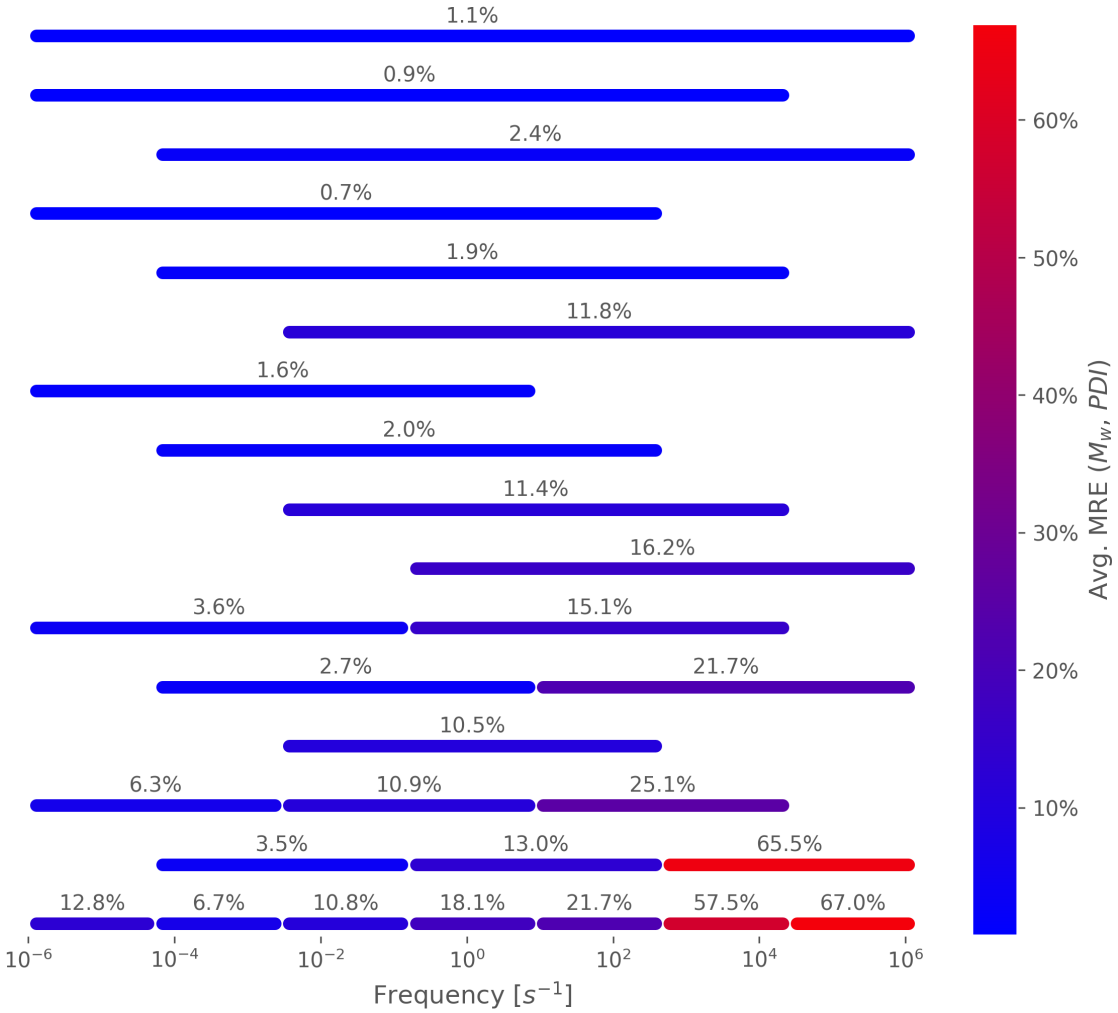

Figure 5.9: Averaged MRE of the M_w and PDI target attributes at various frequency ranges (40,000 training and 10,000 testing instances, unimodal dataset)

The chart shows the prediction performance for the unimodal dataset at the various frequency ranges, with the widest ranges starting at the top and getting smaller towards the bottom. The bars are colour coded based on the average MRE of the two target attributes. The most obvious thing of note in this chart is the substantially worse performance when using only the higher frequency ranges (bottom right of the chart). This confirms our suspicions that the model would not be able to extract enough information from the low variance values at the high frequencies to predict the MWD effectively. In Figure 3.3 we saw that the spread of the data diminishes drastically at around the $10^3 s^{-1}$ mark. This translates perfectly to the bad performance in the last two frequency sectors we see in Figure 5.9. We can also see that not including these frequencies leads to better performance, with the best average MRE being achieved using a frequency range spanning the first five of the seven sectors.

Figure 5.10 shows the same analysis for the bimodal case. Overall the same discoveries also hold here. The last two sectors of the frequency range still see much worse prediction performance than the lower frequency ranges. This time we see the best averaged MRE using a frequency range spanning the first six sectors, however. As this chart shows the predictions of the bimodal model, the errors are also higher across the board than they were in the unimodal case.

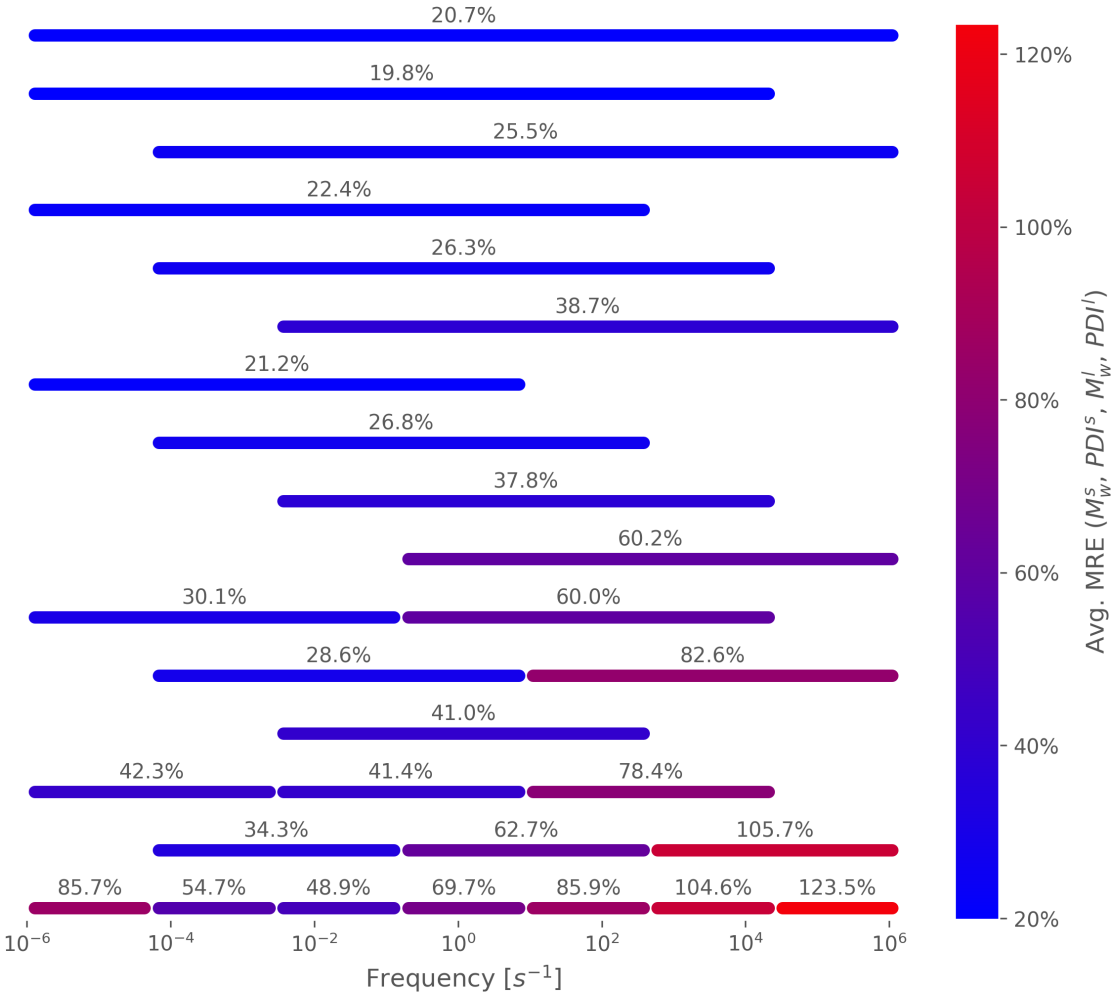

Figure 5.10: Averaged MRE of the M_w^s, PDI^s, M_w^l and PDI^l target attributes at various frequency ranges (180,000 training and 20,000 testing instances, $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

5.5 Proportion of Long-Chain Polymers in the Bimodal Compound

The last thing we wanted to investigate was varying the minimum and maximum proportion of long-chain polymers in the bimodal compounds. More simply put, we tried restricting ϕ^l to various ranges to see how that affected the model's prediction performance. Unaltered, the values of ϕ^l can range from 0 to 1. It stands to reason that the model might struggle to predict the bimodal MWD when ϕ^l is very close to 0 or 1. In this case, one of the two components

of the compound is so dominant that it might be difficult to predict both peaks of the MWD reliably. To test this, we ran a series of tests with various ranges of ϕ^l values. We started with the full range and then decreased the valid range by 0.05 on both sides until reaching a range half the original size. The results of these tests are summarised in Table 5.1.

Table 5.1: Mean relative error (MRE) and the averaged MRE of the M_w^s, PDI^s, M_w^l and PDI^l target attributes, as well as the mean absolute error (MAE) of ϕ^l using various ranges of valid ϕ^l values (180,000 training and 20,000 testing instances, $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

	MRE (M_w^s)	MRE (PDI^s)	MRE (M_w^l)	MRE (PDI^l)	Avg. MRE	MAE (ϕ^l)
$\phi^l \in [0, 1]$	23.40%	22.08%	7.89%	27.32%	20.17%	0.026
$\phi^l \in [0.05, 0.95]$	18.69%	13.59%	4.98%	21.10%	14.59%	0.027
$\phi^l \in [0.1, 0.9]$	16.01%	10.79%	3.78%	13.48%	11.01%	0.026
$\phi^l \in [0.15, 0.85]$	14.01%	11.22%	4.43%	13.26%	10.73%	0.024
$\phi^l \in [0.2, 0.8]$	15.73%	9.68%	3.06%	12.82%	10.32%	0.019
$\phi^l \in [0.25, 0.75]$	13.82%	9.67%	3.25%	10.90%	9.41%	0.023

As we can see, prediction performance tends to improve across all target attributes the more we restrict the range for ϕ^l. In practice, we do not want to restrict the options for ϕ^l too much as we could encounter compounds with small percentages of one component in real life. From the data shown in Table 5.1, it looks like restricting ϕ^l to a range between 0.1 and 0.9 is the sweet spot as this reduces most errors by almost 10% while not being too restrictive. This translates to the compound needing to contain at least 10% of every component.

5.6 Summary of the Best Results

We have gained many insights into how changes to the datasets and their features affect the prediction performance of the models from the previous sections. So we wanted to put it all together and perform one last round of tests using the optimal settings we found during this project. The goal was not to maximise prediction performance at all cost, e.g. restricting ϕ^l to [0.4, 0.6]. Instead, we wanted to apply the setup that made the most sense from a practical standpoint while reducing prediction errors as much as possible.

For the unimodal MWD predictions, we decided to use a training set of 100,000 instances, which is the maximum we could use given the other restrictions we used. As we already collected the data and saw slight performance advantages when using larger datasets, we wanted to use as much of the data as possible. We only used the first 50 features for both G' and G'', as using this frequency range resulted in the best performance in our analysis of frequency ranges. In Section 5.1 we also learned that the performance is substantially worse in the lowest regions of both the target ranges. Therefore, one possibility of increasing the prediction performance is modifying the target ranges to include those lowest values. We generally do not approve

of boosting performance through adjustments such as this, as it is to an extent just moving the goalposts. There are situations where such an approach can be justified, however, e.g. restricting ϕ^l in the bimodal case. It could also make sense if we have specific performance targets to meet that are only achievable by restricting the possible target ranges. We have to keep in mind, however, that the application of the model becomes more restricted as well as it will now only perform as intended in those ranges. For the sake of completeness, we ran the performance tests both with and without restricting the target ranges. For the restricted dataset, we used only the instances where $M_w \geq$ 1,287,000 and $PDI \geq 2$. All the other settings we discussed were applied for both tests. Table 5.2 shows the MAE and MRE metrics for the unimodal neural network using these datasets.

Table 5.2: Mean absolute error (MAE), mean relative error (MRE) and the averaged MRE across all targets (Avg. MRE) of the best performing unimodal models with and without restricting the target ranges by $M_w \geq$ 1,287,000 and $PDI \geq 2$ (100,000 training and 20,000 testing instances, using only the first 50 features each for G' and G'', unimodal dataset)

	MAE (M_w)	MAE (PDI)	MRE (M_w)	MRE (PDI)	Avg. MRE
Full Dataset	8,731	0.018	0.39%	0.98%	0.69%
Restricted Dataset	6,214	0.010	0.10%	0.23%	0.17%

The results shown represent the best prediction performance we achieved using appropriate settings in the unimodal case. As we can see, we managed to drop just below the 0.7% error mark for the average MRE of the M_w and PDI target attributes using the full dataset, which is definitely a very usable prediction performance. We can also see that the prediction performance improved across the board when the lowest regions of the target ranges were excluded. In this test, the model achieved an extremely low average MRE of 0.17%.

For the bimodal MWD predictions, we had a few more things to play with. We used the $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ dataset but extended it by collecting additional data, as we saw that prediction performance still increased with larger training sets in Figure 5.7. As in the unimodal case, we restricted the frequency range of G' and G'' by only including the first 50 features. We also restricted the range of ϕ^l to [0.1, 0.9] as we found this was the sweet spot in Section 5.5. This left us with a dataset of 400,000 instances to feed to the bimodal neural network. As we did in the unimodal case, we also wanted to run a test with restricted target ranges to see how the prediction performance changes. We used the same restrictions of requiring the M_w values to be larger than 1,287,000 and the PDI values to be larger than 2. If we recall the distribution of data points in the target attribute space of the $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ dataset (Figure 3.7), we realise that these conditions are especially restrictive here because the M_w^s values tend to be so small in this dataset. Due to this, we were only left with 45,000 instances for the restricted dataset. The prediction performance for these tests is summarised in Table 5.2.

Table 5.3: Mean relative error (MRE) and the averaged MRE of the M_w^s, PDI^s, M_w^l and PDI^l target attributes, as well as the mean absolute error (MAE) of ϕ^l of the best performing bimodal models with and without restricting the target ranges by $M_w \geq$ 1,287,000 and $PDI \geq$ 2 (360,000 training and 40,000 testing instances for the full dataset, 40,000 training and 5,000 testing instances for the restricted dataset, using only the first 50 features each for G' and G'', $\phi^l \in$ [0.1, 0.9], $\frac{M_w^l}{M_w^s} > PDI_{max}^1$ bimodal dataset)

	MRE (M_w^s)	MRE (PDI^s)	MRE (M_w^l)	MRE (PDI^l)	Avg. MRE	MAE (ϕ^l)
Full Dataset	12.32%	10.53%	3.29%	10.74%	9.22%	0.020
Restricted Dataset	6.83%	5.06%	4.42%	6.78%	5.77%	0.030

As we can see, we were able to push the average MRE below 10% using the full dataset, with the MAE for ϕ^l coming in at 0.02. The other promising takeaway from these results is the MRE of M_w^l that we were able to get to just 3.29%. We have discussed before that this target attribute likely has the most significant influence on a material's physical properties. For the test using restricted target ranges, we can see that most MREs drop significantly, with the average MRE dropping to 5.77%. As we were not able to lower the errors any further using the full dataset, this might be an option to pursue further if sub-5% MREs are required. The restricted dataset results are especially impressive since they were achieved using a training set just $1/9$ the size of the full dataset.

Chapter 6

Summary and Conclusions

The main goal of this project was to investigate the feasibility of using statistical modelling to predict a plastic's MWD from its flow data. We set two problem definitions, one for predicting the two parameters required to describe a unimodal MWD and one for predicting the five parameters required to describe a bimodal MWD.

In the case of unimodal MWDs, we found that the statistical modelling approach is a feasible method to discovering the MWD of a plastic. We managed to achieve MAEs of 8,731 g/mol for the M_w parameter and 0.018 for the PDI parameter using our best performing model, which translated to an average MRE of 0.69%. These are impressive results that should easily be in the realm of acceptable errors for deducing the MWD of a plastic. We also showed that the largest errors were made by our model when at least one of the target attributes was close to the possible minimal value for that attribute. In the rest of the target attribute space, we saw considerably improved prediction performance. When we restricted the dataset to exclude the lowest ranges for the two target variables, we were able to increase the prediction performance considerably to an average MRE of 0.17%. Whether this approach should be used depends on the use case, but as the errors are so small already with the full dataset, using that probably makes more sense for most situations. We also showed that restricting the frequency range of the flow data to a maximum of around $10^3 s^{-1}$ was beneficial for predicting the MWD of polystyrene, as we saw little information in the flow data at higher frequencies. Lastly, we discovered that collecting at least 7,500 instances for training greatly reduces the prediction errors, with training set sizes up to 25,000 instances making sense from a cost-benefit standpoint. Prediction performance can still be improved with larger training sets, but collecting the additional data is probably not worth the minimal performance gains.

The results were less conclusive for the bimodal prediction problem. Here we achieved an average MRE of 9.22% across the M_w^s, PDI^s, M_w^l and PDI^l target attributes and a MAE of

0.02 for ϕ^l with our best model. Whether or not that is good enough will depend on the use case, but the low MRE for M_w^l might help due to the long-chain peak usually dominating the plastic's physical properties. The prediction performance can be improved further by restricting the target ranges to higher values, as we showed in section 5.6. This is due to the significantly higher errors in the lowest regions of the target ranges that we also saw in the unimodal predictions. Applying these restrictions, we were able to lower the average MRE to 5.77%. It is quite likely that this could be pushed even further. However, we could not investigate this as the datasets we collected were not very compatible with this approach due to many of the M_w^s values being in the low regions that would be restricted. When using the datasets we collected without restricting the target range, we showed that collecting at least 40,000 training instances is recommended for the bimodal prediction problem. The performance kept increasing with even larger datasets, so using more data is beneficial if the cost of collecting it is not too high. We also showed that requiring the peaks to be separated has a big performance advantage. We saw the best results by requiring $\frac{M_w^l}{M_w^s} > PDI_{max}^1$. Similarly to the unimodal case, we saw that prediction performance is worst when using only the highest frequency ranges for the flow data. The best result was achieved by restricting the frequency range of G' and G'' to a maximum of around $10^4 s^{-1}$ for the bimodal models, although we also recorded good results by using the same restrictions as in the unimodal case. Finally, we showed that restricting ϕ^l to range from 0.1 to 0.9 is the sweet spot for ensuring good performance while also not restricting the application of the model too much.

In conclusion, for the unimodal prediction problem, the neural network we presented in this project is very capable of finding the MWD of a plastic from its flow data. The statistical modelling approach we showed is, therefore, a feasible alternative to GPC, the standard method of deducing the MWD of a plastic in the industry. Whether it makes sense to use this approach in practice depends on the availability of data to train the model and the quality of the software we used to generate our datasets. Collecting just the 7,500 instances we recommend for training the model using real-world oscillatory shear experiments is likely unachievable due to cost and time constraints. A hybrid approach could work, of course, where real-world data is used in conjunction with software-generated data. In this case, it depends on how accurately the software generates the flow data and whether a model trained on software-generated data could perform equally well on real-world data.

Whether the model presented for the bimodal prediction problem is a feasible alternative to GPC depends on the error tolerance a potential user has. In the form presented in this project, it achieves average MREs around 10%, but this requires a lot of data. We showed that this can be reduced further if we are willing to restrict the possible ranges of the M_w and PDI values. However, we could not thoroughly investigate optimal models and settings for this using the data we collected.

Future work could be carried out to investigate this, using datasets specifically collected with the goal of finding the optimal range of M_w and PDI values to support. Other potential avenues to continue researching the statistical modelling approach to finding MWDs include:

- Collecting much larger datasets and seeing how that affects prediction performance
- Trying to predict MWDs with more than two peaks
- Using different statistical models
- Trying other approaches to separating the peaks in the bimodal case

Bibliography

Ali, S.S., Elsamahy, T., Koutra, E., Kornaros, M., El-Sheekh, M., Abdelkarim, E.A., Zhu, D. and Sun, J., 2021. Degradation of conventional plastic wastes in the environment: a review on current status of knowledge and future perspectives of disposal. *Science of the total environment* [Online], 771, 144719. Available from: https://doi.org/10.1016/j.scitotenv.2020.144719.

Andrady, A.L. and Neal, M.A., 2009. Applications and societal benefits of plastics. *Philosophical transactions of the royal society b: biological sciences* [Online], 364(1526), pp.1977–1984. Available from: https://doi.org/10.1098/rstb.2008.0304.

Anton Paar, 2021. *Basics of rheology* [Online]. Available from: https://wiki.anton-paar.com/en/basics-of-rheology/ [Accessed 25 October 2021].

Balani, K., Verma, V., Agarwal, A. and Narayan, R., 2014. Physical, thermal, and mechanical properties of polymers. In: *Biosurfaces* [Online]. John Wiley & Sons, Ltd, pp.329–344. Available from: https://doi.org/10.1002/9781118950623.app1.

Bower, A.F., 2009. *Applied mechanics of solids* [Online]. CRC Press. Available from: https://doi.org/10.1201/9781439802489.

Chugh, A., 2020. *Mae, mse, rmse, coefficient of determination, adjusted r squared-which metric is better?* [Online]. Medium. Available from: https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e [Accessed 25 November 2021].

Das, C., Inkson, N.J., Read, D.J., Kelmanson, M.A. and McLeish, T.C.B., 2006. Computational linear rheology of general branch-on-branch polymers. *Journal of rheology* [Online], 50(2), pp.207–234. Available from: https://doi.org/10.1122/1.2167487.

Dealy, J.M., Read, D.J. and Larson, R.G., 2018. *Structure and rheology of molten polymers: from structure to flow behavior and back again* [Online]. 2^{nd} ed. Hanser. Available from: https://doi.org/10.3139/9781569906125.

Eriksen, M., Lebreton, L.C.M., Carson, H.S., Thiel, M., Moore, C.J., Borerro, J.C., Galgani, F., Ryan, P.G. and Reisser, J., 2014. Plastic pollution in the world's oceans: more than 5 trillion plastic pieces weighing over 250,000 tons afloat at sea. *Plos one* [Online], 9(12), pp.1–15. Available from: https://doi.org/10.1371/journal.pone.0111913.

Generalic, E., 2018. *Styrene* [Online]. Croatian-English Chemistry Dictionary & Glossary. Available from: https://glossary.periodni.com/glossary.php?en=styrene [Accessed 21 October 2021].

Geyer, R., Jambeck, J.R. and Law, K.L., 2017. Production, use, and fate of all plastics ever made. *Science advances* [Online], 3(7), e1700782. Available from: https://doi.org/10.1126/sciadv.1700782.

Helmenstine, A.M., 2018. *Relative error definition (science)* [Online]. ThoughtCo. Available from: https://www.thoughtco.com/definition-of-relative-error-605609 [Accessed 25 November 2021].

Hopewell, J., Dvorak, R. and Kosior, E., 2009. Plastics recycling: challenges and opportunities. *Philosophical transactions of the royal society b: biological sciences* [Online], 364(1526), pp.2115–2126. Available from: https://doi.org/10.1098/rstb.2008.0311.

Hosler, D., Burkett, S.L. and Tarkanian, M.J., 1999. Prehistoric polymers: rubber processing in ancient mesoamerica. *Science* [Online], 284(5422), pp.1988–1991. Available from: https://doi.org/10.1126/science.284.5422.1988.

Impact Plastics, 2017. *The difference between amorphous & semi-crystalline polymers* [Online]. Available from: https://blog.impactplastics.co/blog/the-difference-between-amorphous-semi-crystalline-polymers [Accessed 23 October 2021].

Jansen, J.A., 2016. *Plastics – it's all about molecular structure* [Online]. The Madison Group. Available from: https://www.madisongroup.com/publications/PE%20Sept%20-%20Consultants%20Corner.pdf [Accessed 21 October 2021].

Meyers, M.A. and Chawla, K.K., 2008. *Mechanical behavior of materials* [Online]. 2^{nd} ed. Cambridge University Press. Available from: https://doi.org/10.1017/CBO9780511810947.

Science History Institute, 2019. *Science of plastics* [Online]. Available from: https://www.sciencehistory.org/science-of-plastics [Accessed 21 October 2021].

Appendix A

Supplementary Figures

A.1 Data Exploration

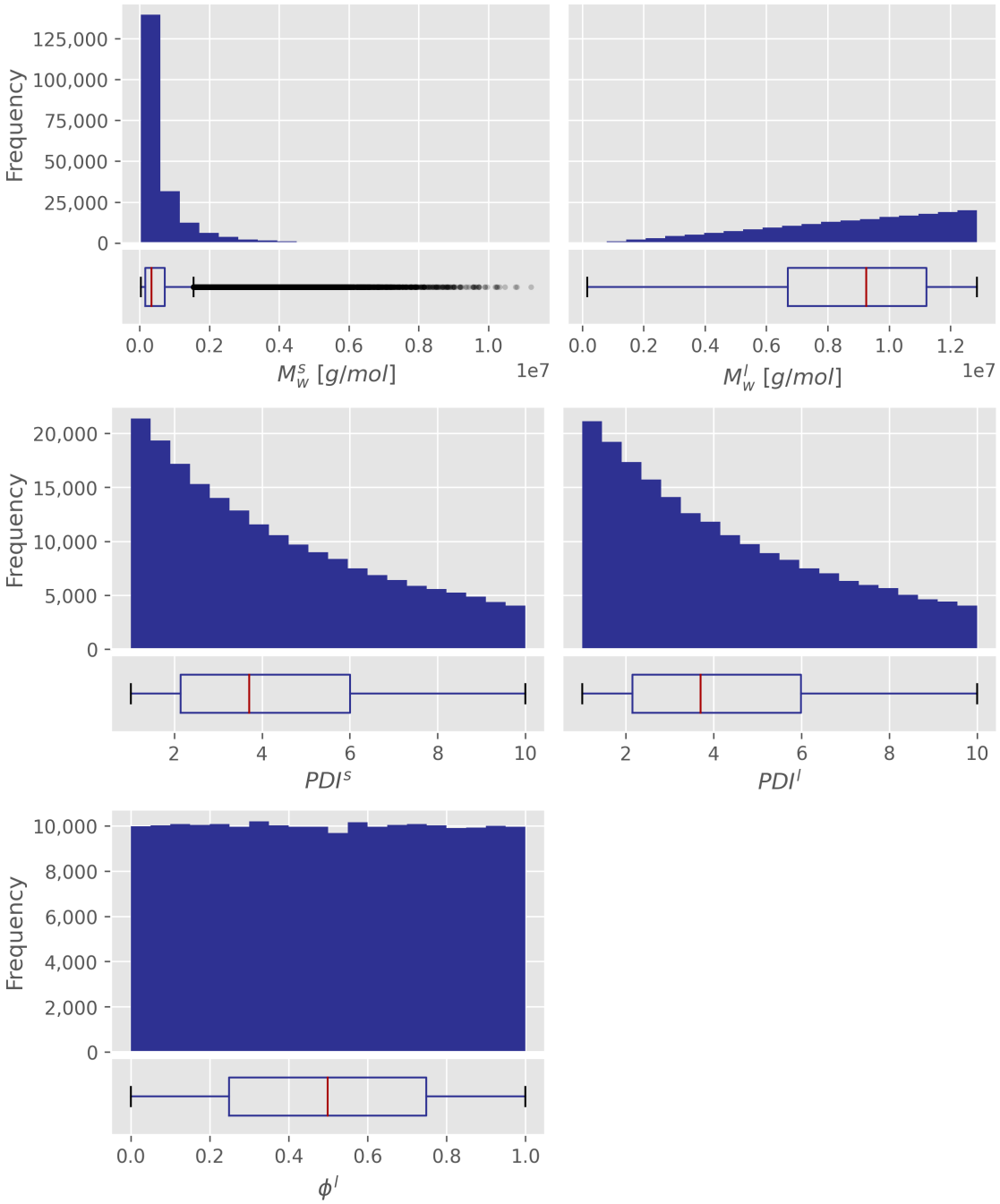

Figure A.1: Histograms and boxplots of the M_w^s, PDI^s, M_w^l, PDI^l and ϕ^l target attributes ($\frac{M_w^l}{M_w^s} > PDI_{max}^{1.5}$ bimodal dataset)

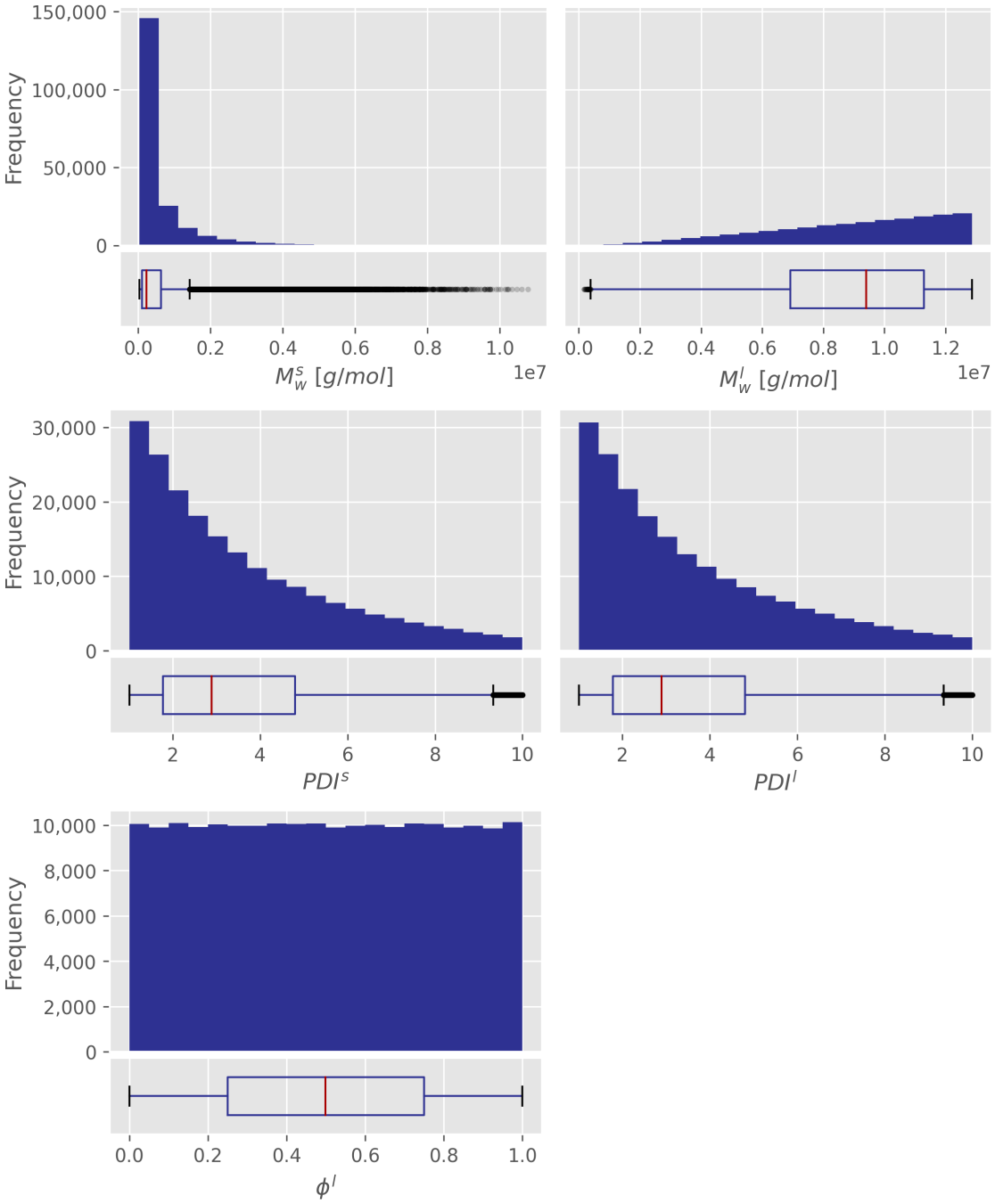

Figure A.2: Histograms and boxplots of the M_w^s, PDI^s, M_w^l, PDI^l and ϕ^l target attributes ($\frac{M_w^l}{M_w^s} > PDI_{max}^2$ bimodal dataset)

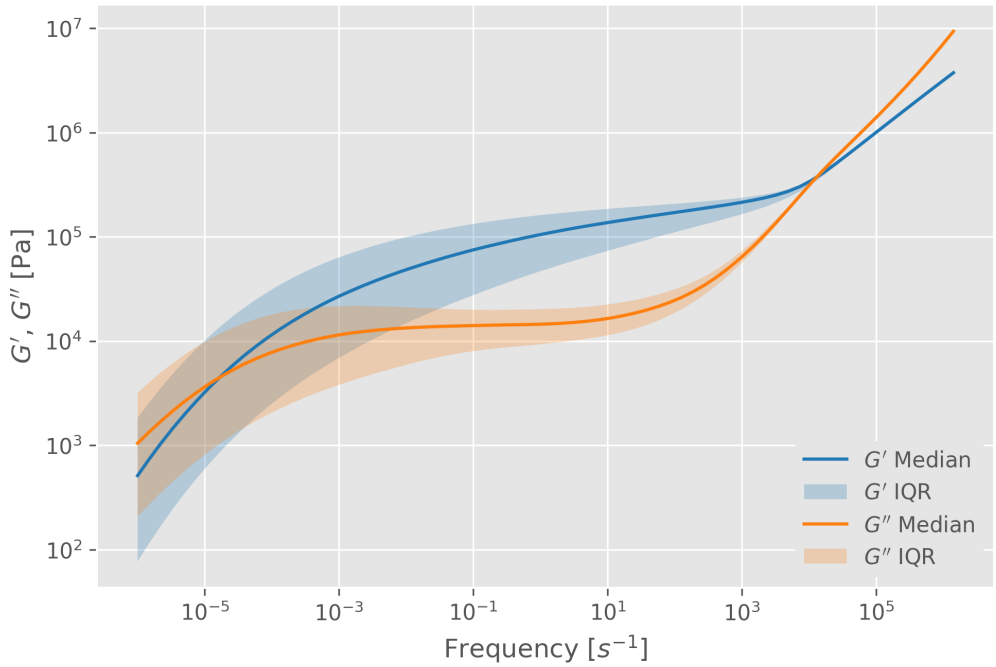

Figure A.3: Median and IQR of the storage- and loss modulus curves ($\frac{M_w^l}{M_w^s} > PDI_{max}^{1.5}$ bimodal dataset)

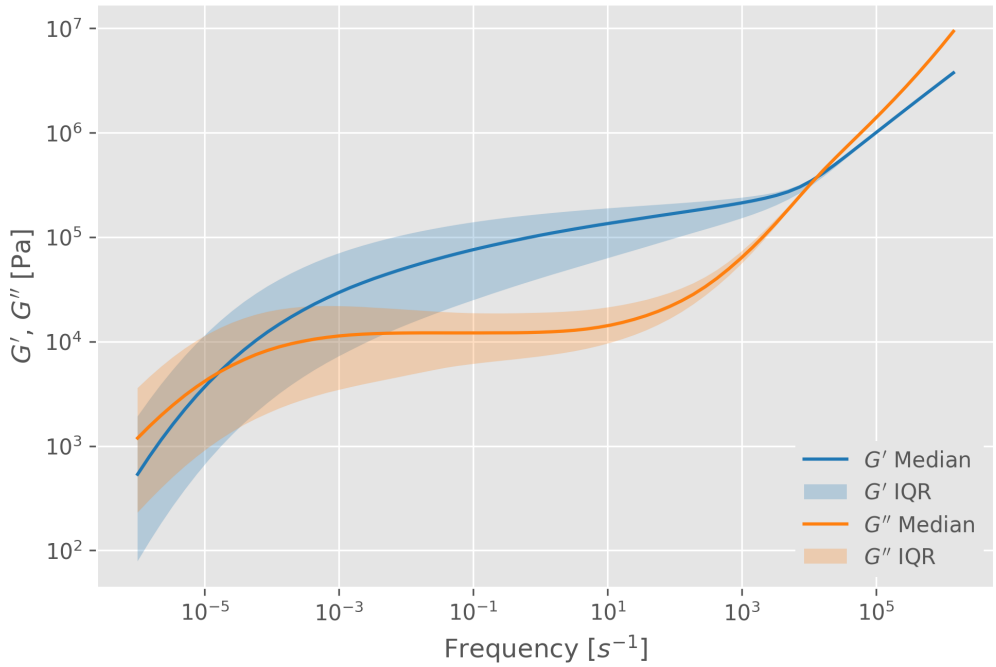

Figure A.4: Median and IQR of the storage- and loss modulus curves ($\frac{M_w^l}{M_w^s} > PDI_{max}^{2}$ bimodal dataset)

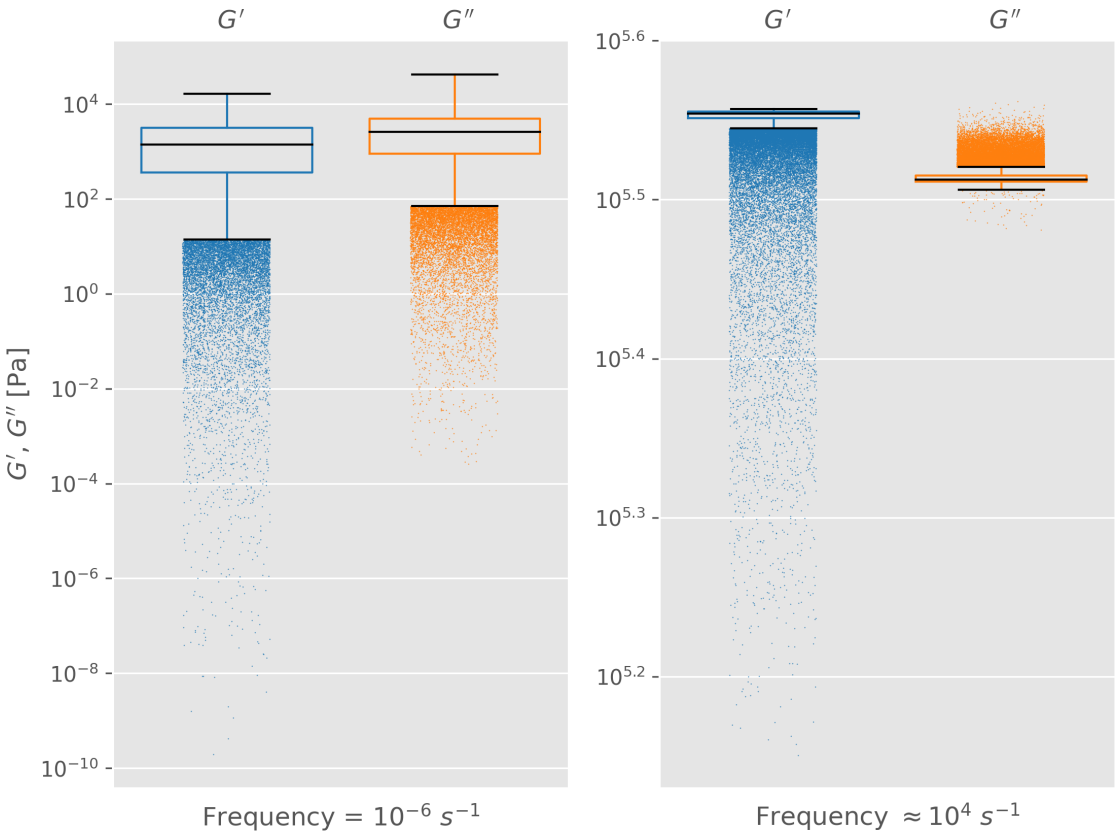

Figure A.5: Boxplots showing the distribution of G' and G'' at the lowest frequency and at a high frequency (no restrictions bimodal dataset)

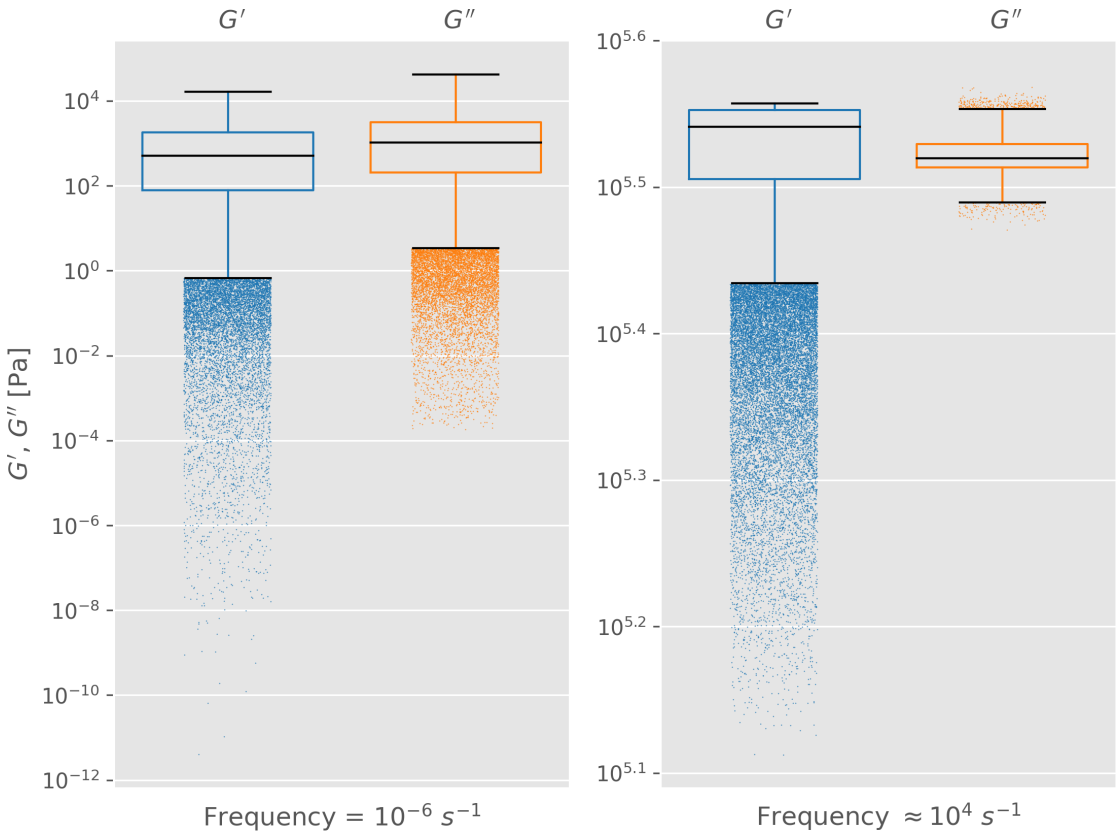

Figure A.6: Boxplots showing the distribution of G' and G'' at the lowest frequency and at a high frequency ($\frac{M_w^l}{M_w^s} > PDI_{max}^{1.5}$ bimodal dataset)

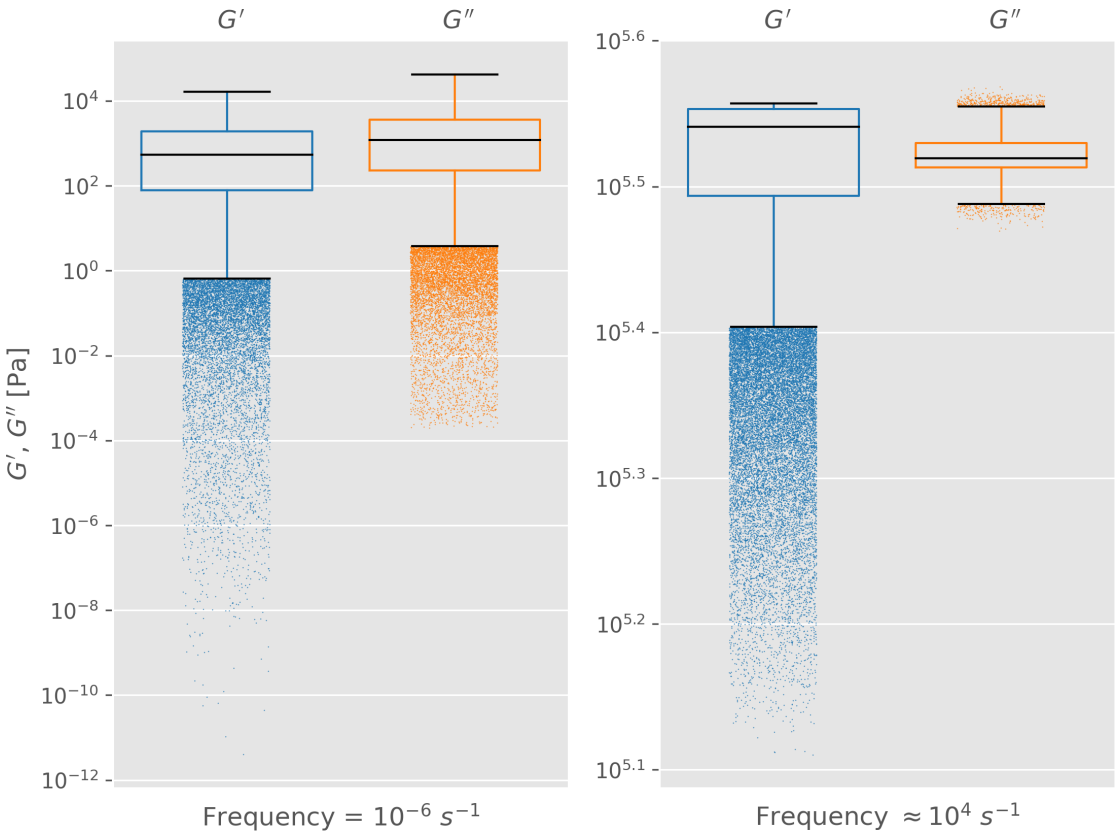

Figure A.7: Boxplots showing the distribution of G' and G'' at the lowest frequency and at a high frequency ($\frac{M_w^l}{M_w^s} > PDI_{max}^2$ bimodal dataset)

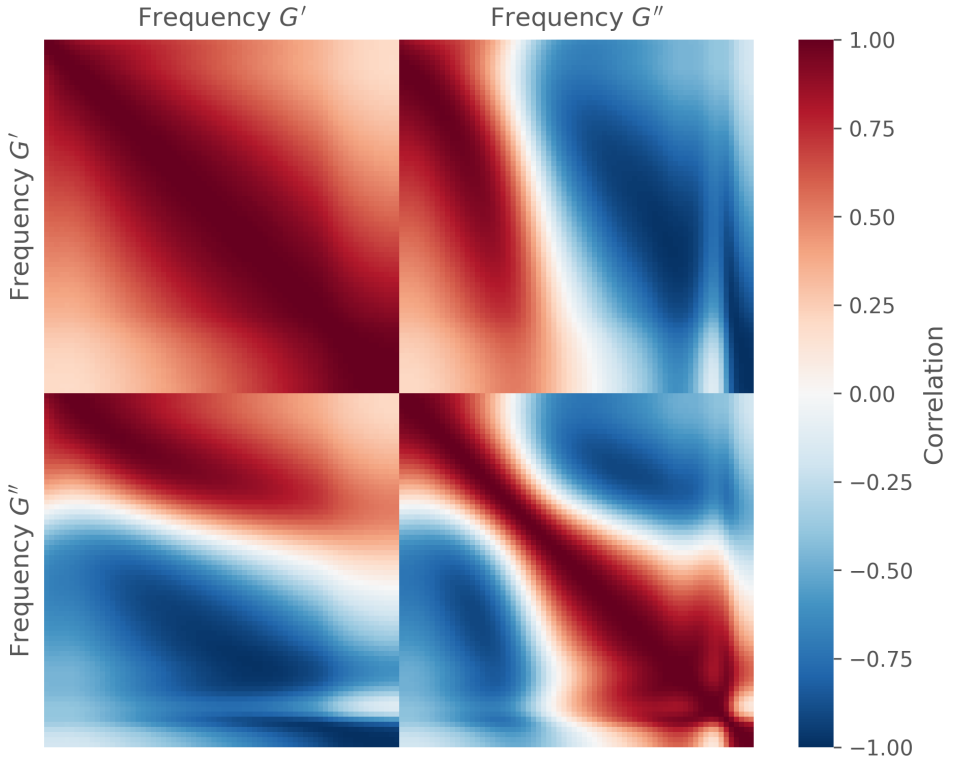

Figure A.8: Correlation matrix of all features (no restrictions bimodal dataset)

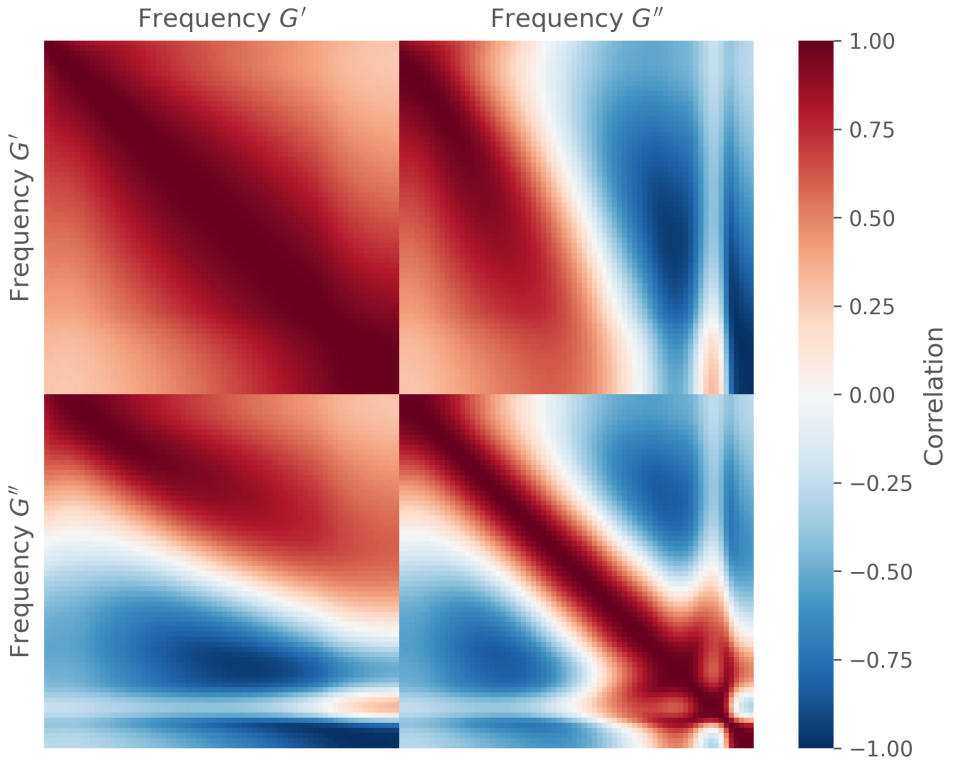

Figure A.9: Correlation matrix of all features ($\frac{M_w^l}{M_w^s} > PDI_{max}^{1.5}$ bimodal dataset)

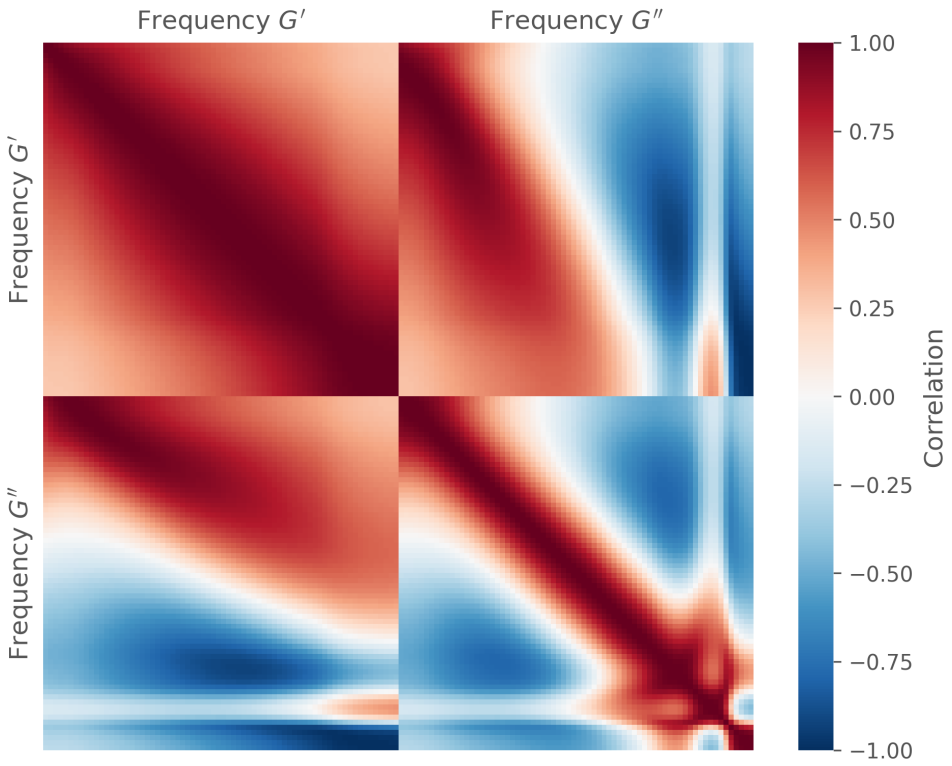

Figure A.10: Correlation matrix of all features ($\frac{M_w^l}{M_w^s} > PDI_{max}^2$ bimodal dataset)

A.2 Model Performance

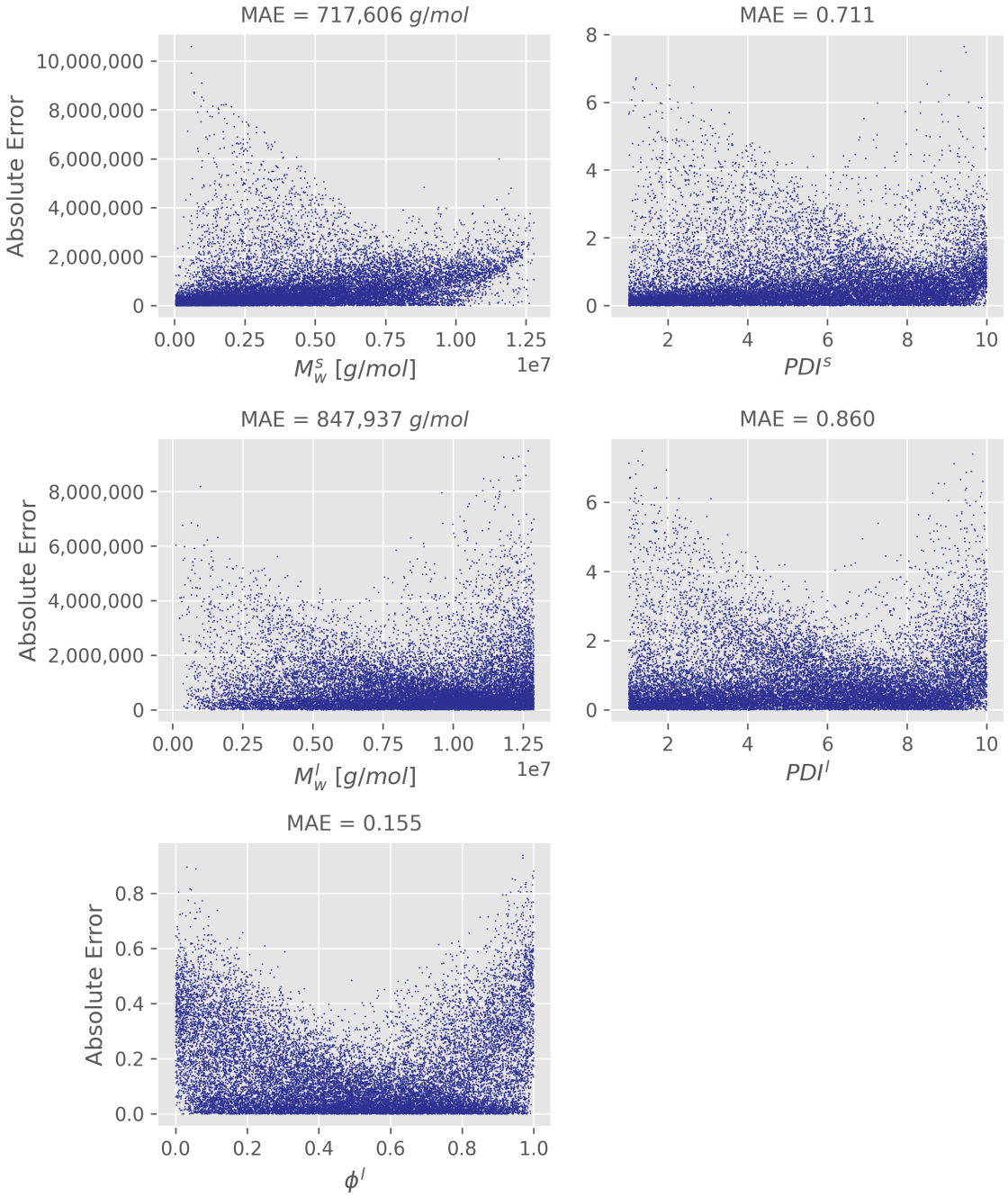

Figure A.11: Absolute errors of M_w^s, PDI^s, M_w^l, PDI^l and ϕ^l across their true value range (180,000 training and 20,000 testing instances, no restrictions bimodal dataset)

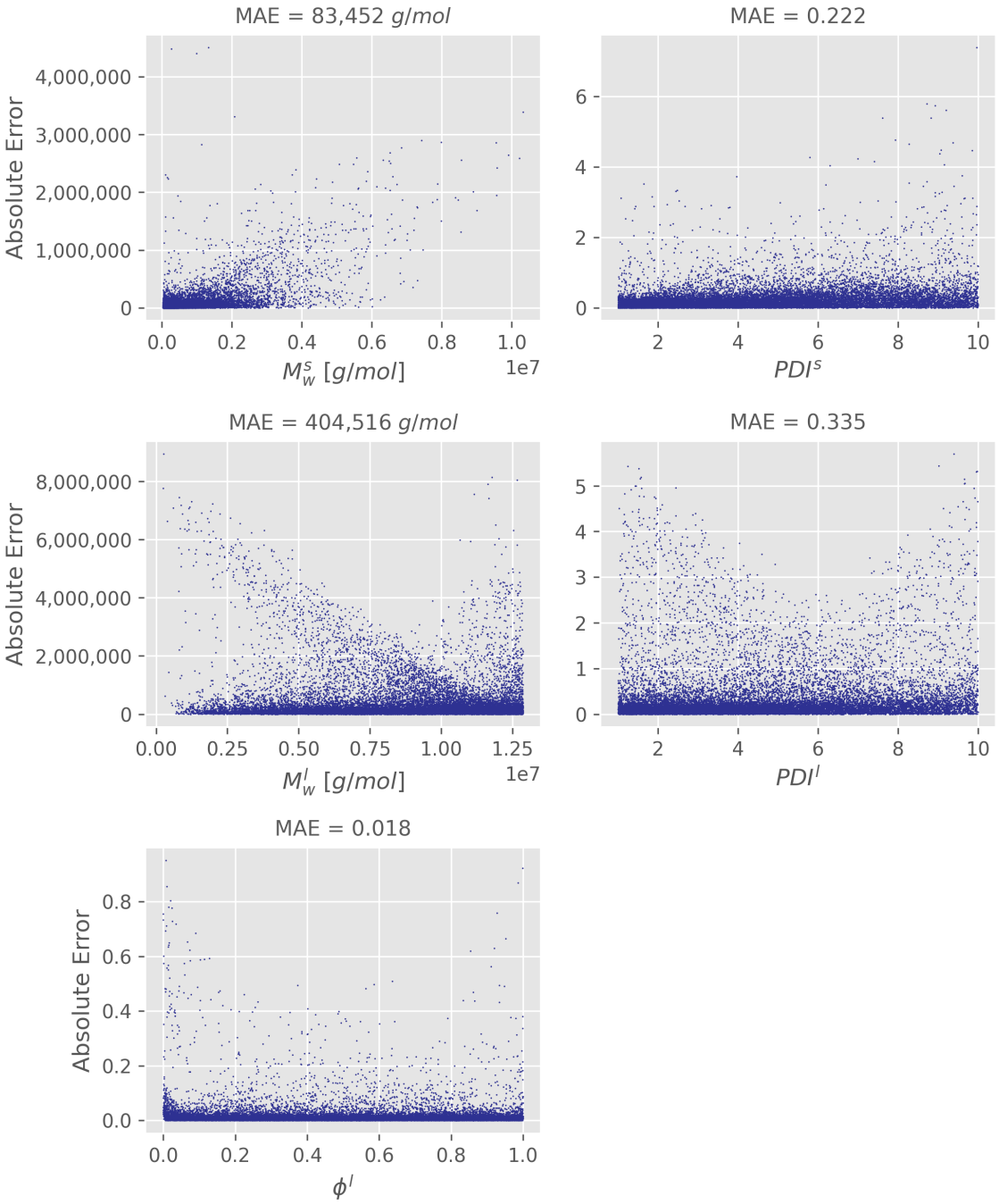

Figure A.12: Absolute errors of M_w^s, PDI^s, M_w^l, PDI^l and ϕ^l across their true value range (180,000 training and 20,000 testing instances, $\frac{M_w^l}{M_w^s} > PDI_{max}^{1.5}$ bimodal dataset)

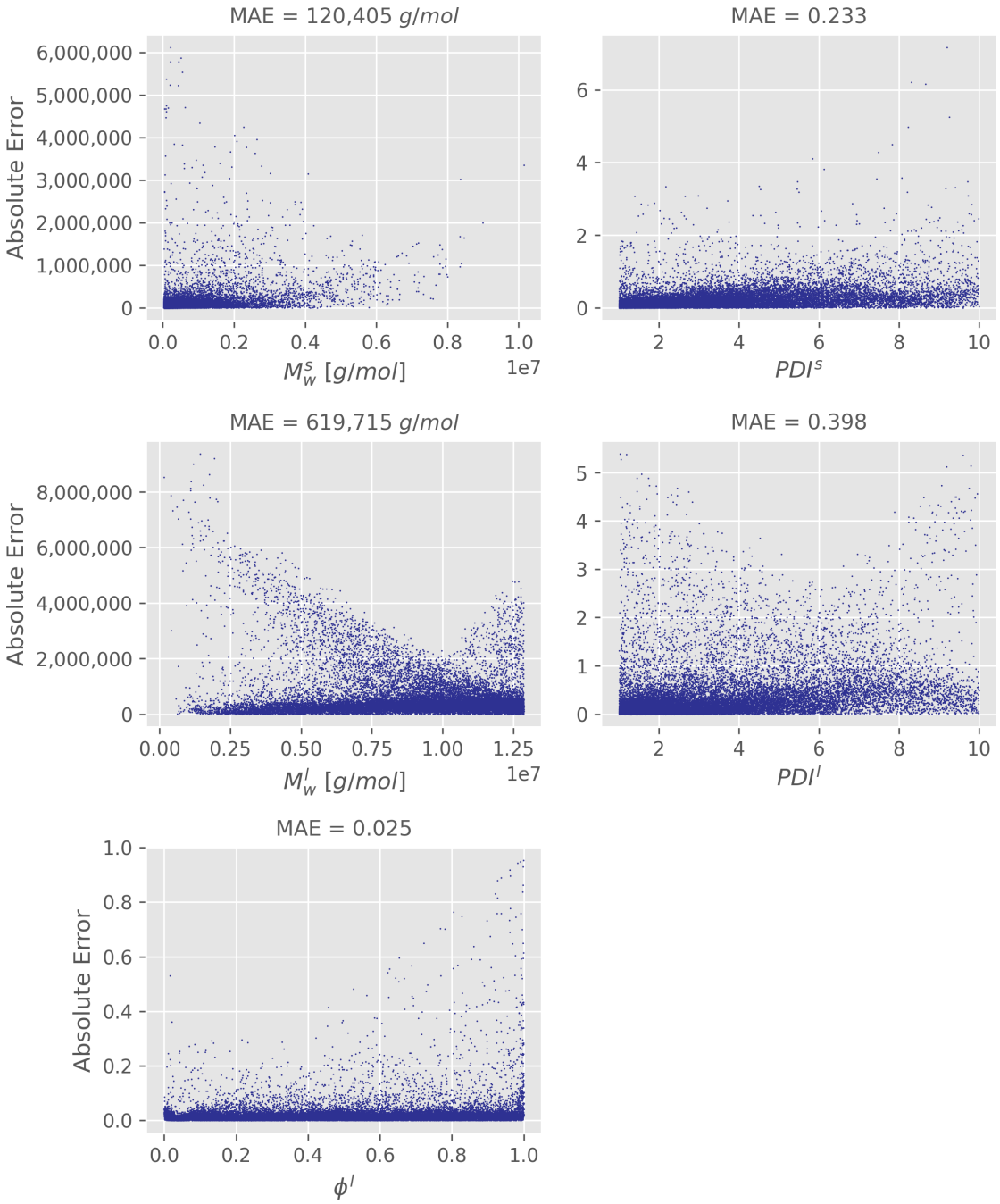

Figure A.13: Absolute errors of M_w^s, PDI^s, M_w^l, PDI^l and ϕ^l across their true value range (180,000 training and 20,000 testing instances, $\frac{M_w^l}{M_w^s} > PDI_{max}^2$ bimodal dataset)

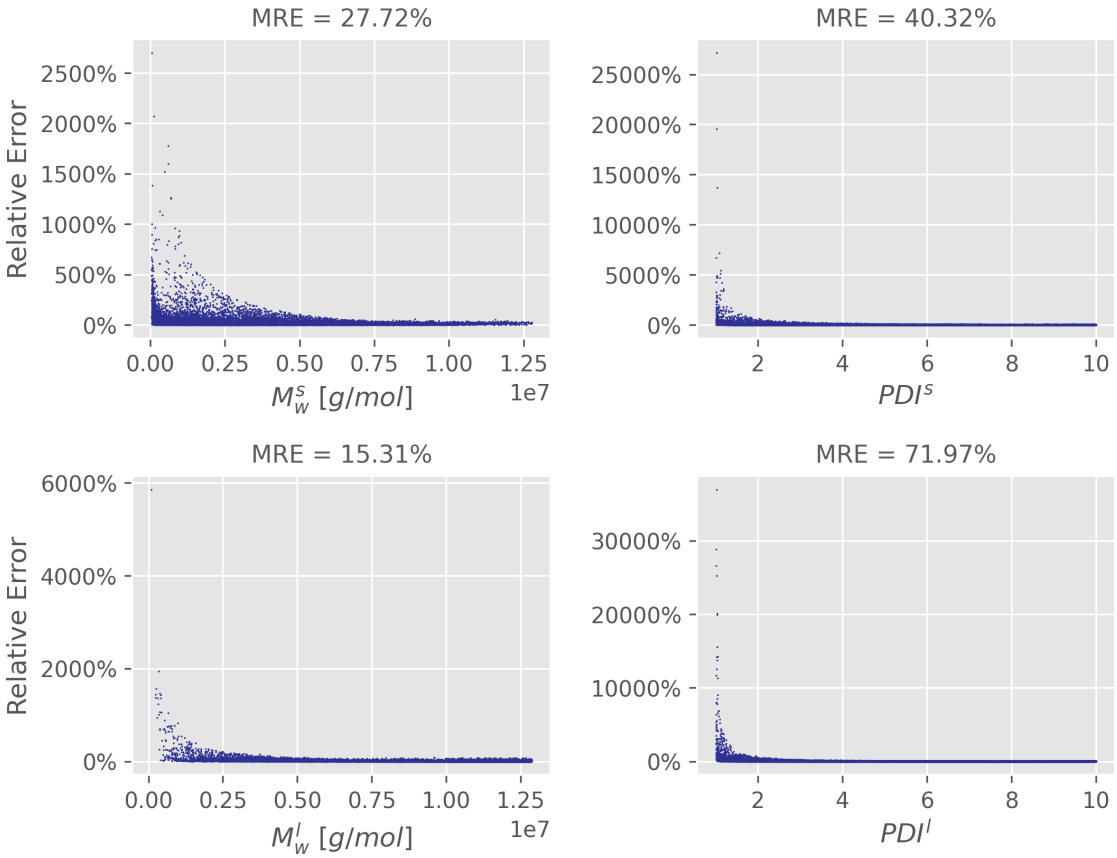

Figure A.14: Relative errors of M_w^s, PDI^s, M_w^l and PDI^l across their true value range (180,000 training and 20,000 testing instances, no restrictions bimodal dataset)

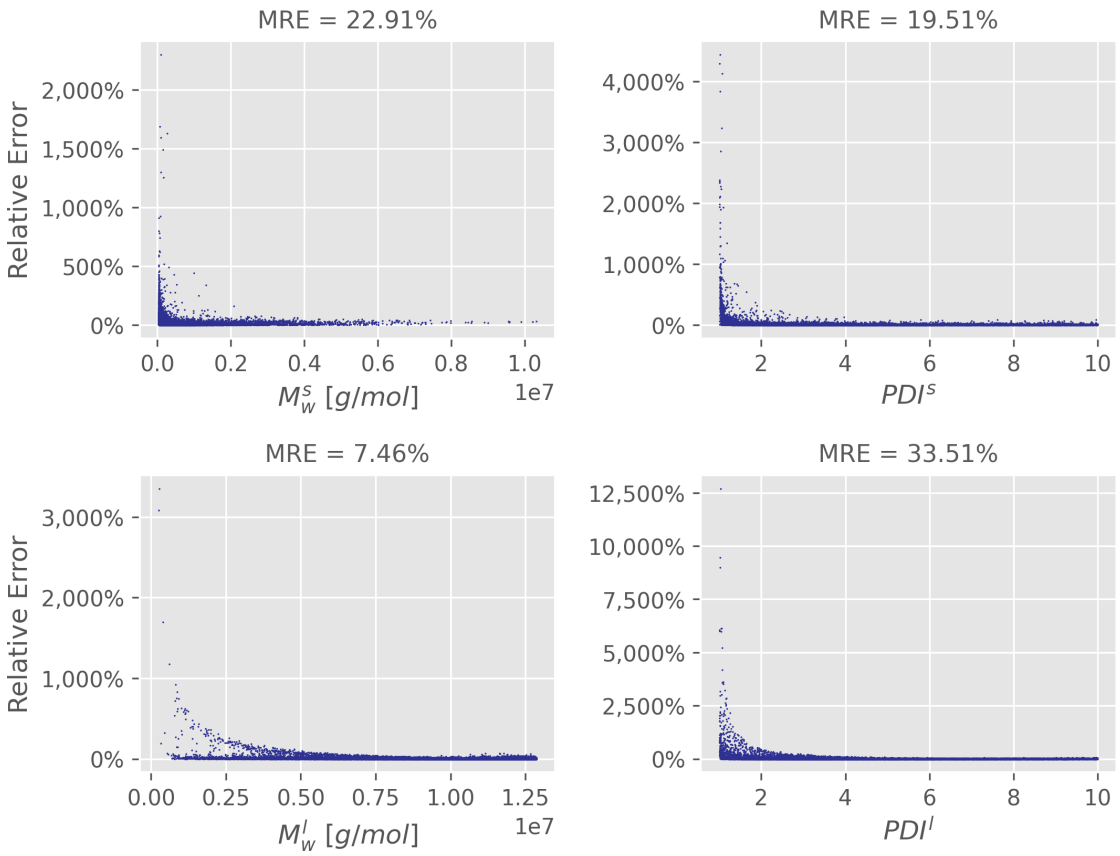

Figure A.15: Relative errors of M_w^s, PDI^s, M_w^l and PDI^l across their true value range (180,000 training and 20,000 testing instances, $\frac{M_w^l}{M_w^s} > PDI_{max}^{1.5}$ bimodal dataset)

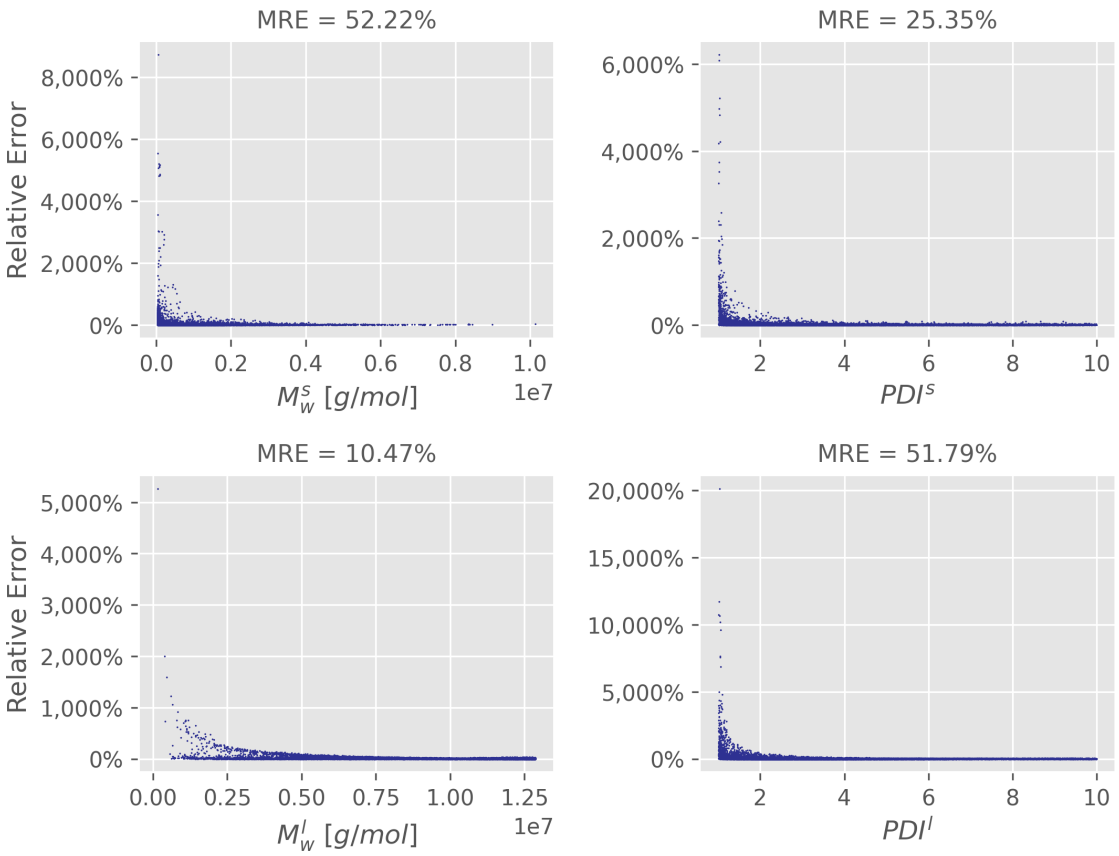

Figure A.16: Relative errors of M_w^s, PDI^s, M_w^l and PDI^l across their true value range (180,000 training and 20,000 testing instances, $\frac{M_w^l}{M_w^s} > PDI_{max}^2$ bimodal dataset)

A.3 Training Set Size

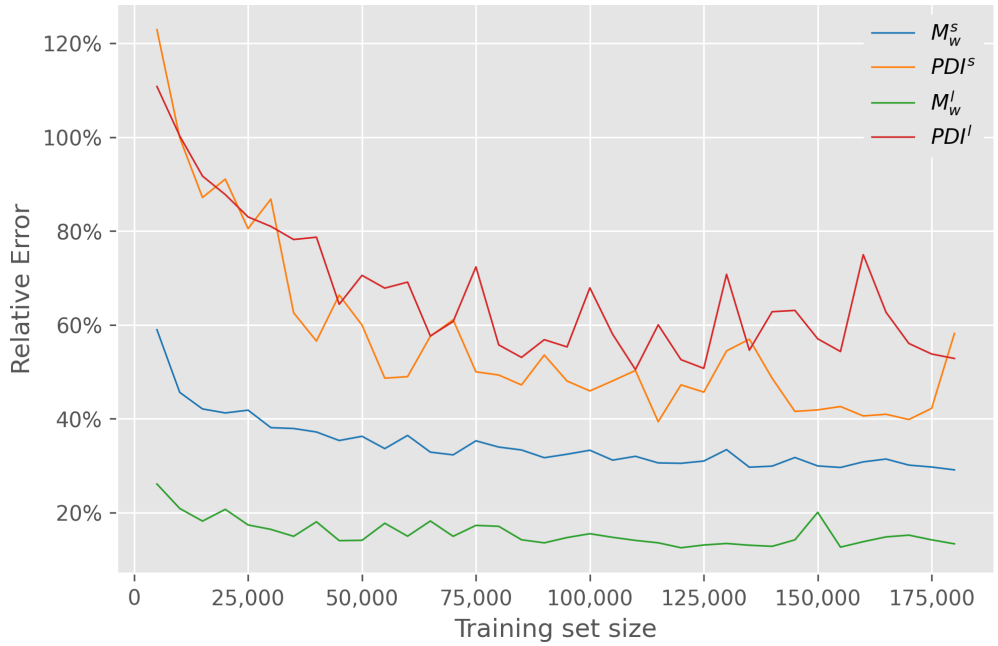

Figure A.17: Mean relative error for the target attributes M_w^s, PDI^s, M_w^l and PDI^l across various sizes of training sets (20,000 testing instances, no restrictions bimodal dataset)

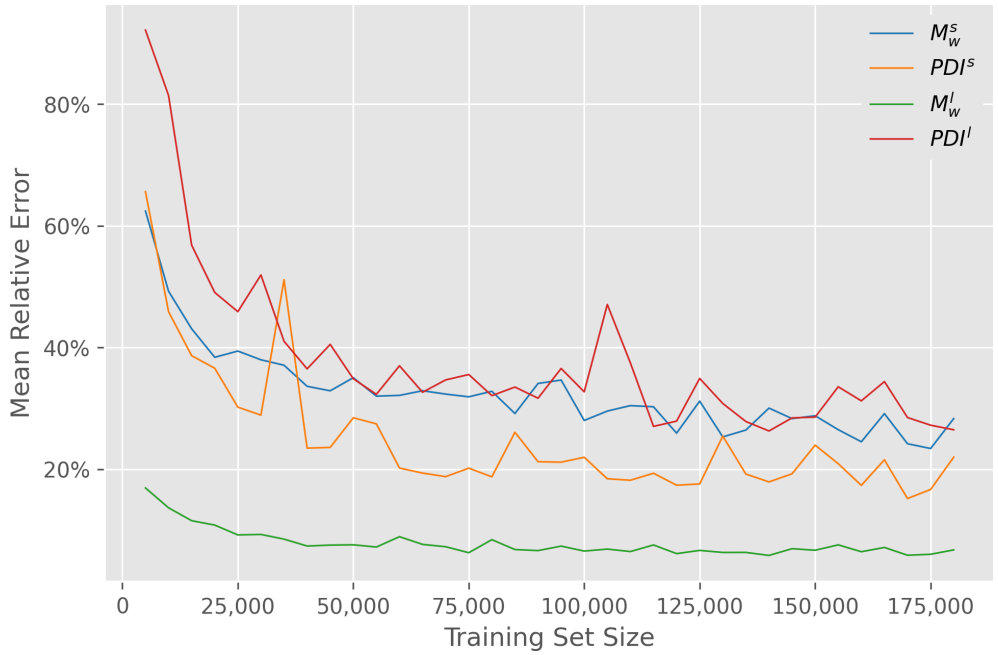

Figure A.18: Mean relative error for the target attributes M_w^s, PDI^s, M_w^l and PDI^l across various sizes of training sets (20,000 testing instances, $\frac{M_w^l}{M_w^s} > PDI_{max}^{1.5}$ bimodal dataset)

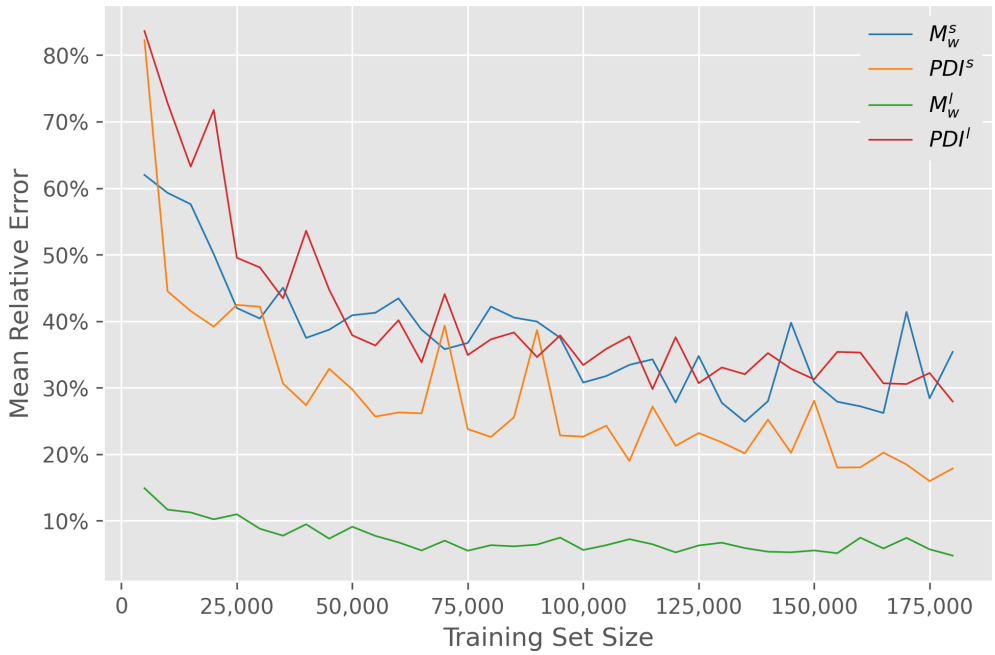

Figure A.19: Mean relative error for the target attributes M_w^s, PDI^s, M_w^l and PDI^l across various sizes of training sets (20,000 testing instances, $\frac{M_w^l}{M_w^s} > PDI_{max}^2$ bimodal dataset)

Appendix B

Supplementary Tables

Table B.1: Summary statistics for the M_w^s, PDI^s, M_w^l, PDI^l, and ϕ^l target attributes ($\frac{M_w^l}{M_w^s} > PDI_{max}^{1.5}$ bimodal dataset)

	M_w^s [g/mol]	PDI^s	M_w^l [g/mol]	PDI^l	ϕ^l
count	200,000	200,000	200,000	200,000	200,000
unique count	180,379	200,000	197,868	200,000	200,000
mean	633,286	4.25	8,754,026	4.24	0.499
std	856,253	2.45	2,914,835	2.44	0.289
min	38,610	1.01	154,570	1.01	0.000
1st quartile	165,759	2.15	6,700,995	2.15	0.249
median	338,116	3.71	9,256,472	3.70	0.498
3rd quartile	718,726	6.00	11,218,703	5.99	0.749
max	11,217,065	10.00	12,869,922	10.00	1.000

Table B.2: Summary statistics for the M_w^s, PDI^s, M_w^l, PDI^l, and ϕ^l target attributes ($\frac{M_w^l}{M_w^s} > PDI_{max}^{2}$ bimodal dataset)

	M_w^s [g/mol]	PDI^s	M_w^l [g/mol]	PDI^l	ϕ^l
count	200,000	200,000	200,000	200,000	200,000
unique count	172,018	200,000	197,750	200,000	200,000
mean	579,647	3.54	8,896,587	3.54	0.500
std	900,087	2.18	2,832,515	2.18	0.289
min	38,611	1.01	161,873	1.01	0.000
1st quartile	104,860	1.78	6,924,306	1.78	0.250
median	235,328	2.89	9,398,128	2.90	0.499
3rd quartile	635,669	4.80	11,286,412	4.81	0.750
max	10,775,482	10.00	12,869,952	10.00	1.000

Table B.3: Summary statistics for the G' and G'' features (no restrictions bimodal dataset)

	G' [Pa]	G'' [Pa]
count	14,000,000	14,000,000
unique count	6,562,620	5,890,578
mean	3.75×10^5	5.33×10^5
std	6.96×10^5	1.56×10^6
min	1.97×10^{-10}	2.62×10^{-4}
1st quartile	5.14×10^4	1.39×10^4
median	1.71×10^5	2.38×10^4
3rd quartile	2.45×10^5	7.05×10^4
max	3.75×10^6	9.53×10^6

Table B.4: Summary statistics for the G' and G'' features ($\frac{M_w^l}{M_w^s} > PDI_{max}^{1.5}$ bimodal dataset)

	G' [Pa]	G'' [Pa]
count	14,000,000	14,000,000
unique count	8,736,841	8,018,414
mean	3.44×10^5	5.32×10^5
std	7.00×10^5	1.56×10^6
min	4.08×10^{-12}	1.81×10^{-4}
1st quartile	1.71×10^4	8.30×10^3
median	1.09×10^5	2.01×10^4
3rd quartile	2.27×10^5	8.14×10^4
max	3.75×10^6	9.55×10^6

Table B.5: Summary statistics for the G' and G'' features ($\frac{M_w^l}{M_w^s} > PDI_{max}^{2}$ bimodal dataset)

	G' [Pa]	G'' [Pa]
count	14,000,000	14,000,000
unique count	8,776,339	8,276,437
mean	3.43×10^5	5.32×10^5
std	6.99×10^5	1.56×10^6
min	4.08×10^{-12}	1.99×10^{-4}
1st quartile	1.68×10^4	7.54×10^3
median	1.08×10^5	1.94×10^4
3rd quartile	2.26×10^5	8.20×10^4
max	3.75×10^6	9.56×10^6

Appendix C

Code

All the scripts, jupyter notebooks, and code used to generate the charts and train the neural networks can be found on the GitHub repository for this project, available at:

https://github.com/bjeffr/mwd_prediction