

# Melt Instability Identification Using Unsupervised Machine Learning Algorithms

Alex Gansen, Julian Hennicker, Clemens Sill, Jean Dheur, Jack S. Hale, and Jörg Baller\*

In industrial extrusion processes, increasing shear rates can lead to higher production rates. However, at high shear rates, extruded polymers and polymer compounds often exhibit melt instabilities ranging from stick-slip to sharkskin to gross melt fracture. These instabilities result in challenges to meet the specifications on the extrudate shape. Starting with an existing published data set on melt instabilities in polymer extrusion, we assess the suitability of clustering, unsupervised machine learning algorithms combined with feature selection, to extract and identify hidden and important features from this data set, and their possible relationship with melt instabilities. The data set consists of both intrinsic features of the polymer as well as extrinsic features controlled and measured during an extrusion experiment. Using a range of commonly available clustering algorithms, it is demonstrated that the features related to only the intrinsic properties of the data set can be reliably divided into two clusters, and that in turn, these two clusters may be associated with either the stick-slip or sharkskin instability. Furthermore, using a feature ranking on both the intrinsic and extrinsic features of the data set, it is shown that the intrinsic properties of molecular weight and polydispersity are the strongest indicators of clustering.

## 1. Introduction

Melt instabilities are a critical factor limiting the maximum throughput of industrial extrusion processes. These melt instabilities appear with increasing shear rate. For a typical polymer undergoing increasing extrusion shear rates, one expects to see the sharkskin instability at low shear rates, followed by a transition to the stick-slip regime, and finally gross-melt-fracture.

The sharkskin instability is a surface instability of height far smaller than the thickness of the extrudate. When sharkskin is well developed it manifests as a periodic structure with amplitude of a few tens to hundreds of microns over the whole extruded sample surface. Although the presence of sharkskin does not necessarily alter the physical properties of the bulk extrudate, it does lead to a change in the surface texture of the extrudate. Sharkskin also turns thin films, commonly used in packaging industry from transparent to opaque due to the light scattering on the rough


surface. The stick-slip instability, referred to as just stick-slip henceforth, is characterized by alternating smooth and rough regions at the extrudates surface. It is accompanied by important pressure fluctuations of about 10% of the mean pressure measured by the pressure transducer in the main barrel of the capillary rheometer. The gross-melt instability, is characterized by the distortion of the whole extrudate. For a full review of melt instabilities, the interested reader is referred to refs. [1, 2].

In the present work, the focus lies on the sharkskin and stick slip melt instability. They are particularly interesting as many polymers only show one of these instabilities. A given instability is influenced by the intrinsic properties of the polymer like polydispersity, molecular weight, branching etc.<sup>[3]</sup> and external factors like temperature, interaction at the polymer/wall interface, shape of die, and die entry. To investigate the impact of the different intrinsic properties, machine learning (ML) techniques can be applied. Many ML ideas and techniques like multilayered neural networks,<sup>[4,5]</sup> backpropagation algorithms,<sup>[6-8]</sup> etc. have already been developed starting from the 60's. However, only the significant increase of computational resources like data storage and increasing computational performance combined with many open source libraries like for example, TensorFlow<sup>[9]</sup> and scikit-learn<sup>[10]</sup> allowed the application of these techniques to the challenging problems of today. Generally, ML techniques can be split

A. Gansen, J. Baller  
Department of Physics and Materials Science  
University of Luxembourg  
162A Avenue de la Faiencerie, Luxembourg L-1511, Grand Duchy of Luxembourg  
E-mail: joerg.baller@uni.lu

J. Hennicker, J. S. Hale  
Department of Engineering  
University of Luxembourg  
Maison du Nombre 6, Avenue de la Fonte, Esch-sur-Alzette L-4364,  
Grand Duchy of Luxembourg

C. Sill, J. Dheur  
Goodyear Innovation Center Luxembourg  
Avenue Gordon Smith, Colmar-Berg L-7750, Grand Duchy of Luxembourg

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/mame.202200628>

© 2023 The Authors. Macromolecular Materials and Engineering published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/mame.202200628



in two groups, supervised and unsupervised learning.<sup>[11]</sup> In supervised learning, the algorithm learns on a labeled data set and the ML algorithm uses the label (output) to adjust the model parameters. Unsupervised techniques on the other hand use unlabeled data, the algorithms look for similarities inside the data and group them accordingly.

Focusing on the field of rheology, there exist many publications applying supervised learning approaches like regressions to predict the viscosity of materials for example refs. [12–14]. Lie et al.<sup>[15]</sup> on the other hand developed a predictive multiscale method that allows them to directly predict the viscoelastic properties like dynamic moduli and zero-shear rate viscosity of polymers. Tariq et al. used artificial neural networks to predict rheological parameters of cement with nanoclay.<sup>[16]</sup>

Saad<sup>[17]</sup> used supervised and unsupervised machine learning techniques to study binary compounds and to predict their crystal structure and then associate them to a specific class, like photovoltaic, superconductors, and ferromagnetic materials. Contrary to this work, they applied dimensionality reduction techniques whereas here the focus lies on clustering algorithms. Abbassi et al.<sup>[18]</sup> apply unsupervised data classification to data sets from nanoelectronics and spectroscopy to identify meaningful structures in data sets. In medicine, unsupervised machine learning methods are for example used to cluster patients based on their genomic makeup without providing input parameters a priori.<sup>[19]</sup> Another example would be the application of clustering algorithms to separate between high energy events in particle physics.<sup>[20]</sup> Only recently, supervised,<sup>[21]</sup> unsupervised,<sup>[22]</sup> as well as a combination of both ML techniques<sup>[23]</sup> have been applied to the identification of phases and the prediction of phase change behavior in polymers and polymer compounds.

However, focusing on unsupervised learning algorithms, especially clustering, to our knowledge no publications in the field of polymer rheology could be found. This study is based on the publication in ref. [3]. All the data which are used in the present work that apply the different ML techniques are taken from this publication. No additional measurements have been carried out. Filipe et al. investigated the effect of polydispersity, molecular weight, and topology (branching) on the melt instability. To understand the effect of the intrinsic properties of a polymer on the melt instability, an unsupervised machine learning technique, referred to as clustering is used. In this paper, the aim is to establish a link between a cluster and corresponding melt instability. Therefore, nine different clustering algorithms are applied. To understand which features (molecular weight, branching, polydispersity, etc.) have the strongest impact on the clustering, a variety of feature selection algorithms are used, once the sample has been clustered. This results in a hierarchical ranking of the features, with the features having the strongest impact on the clustering being ranked first.

## 2. Data and Method

### 2.1. Supervised versus Unsupervised Learning

Machine learning algorithms can be split in two main groups, supervised and unsupervised learning algorithms. Supervised algorithms can further be divided in regression and classification tasks whereas unsupervised algorithms can be divided in dimen-

sionality reduction and clustering. To illustrate the difference between a supervised and unsupervised problem a classification task is compared to a clustering task. For a classification task, the inputs are for example the images of melt instabilities like sharkskin, stick-slip, or gross melt fracture with the corresponding label, as illustrated in the top box in Figure 1a. A supervised training algorithm is then trained on these images and labels. After successful training, the classification algorithm should be able to correctly label an unlabeled image (bottom box in Figure 1a). A clustering algorithm on the other hand only receives pictures (or other data as input) without any label. The algorithm should then group the images in corresponding clusters based on their similarity. The shown unsupervised example corresponds exactly to what we intend to do, except that we do not use pictures of the extrudate but rather information about the polymer topology to cluster polymers with similar properties in the same cluster which then can be associated to a melt instability. In our case, the algorithms should create clusters where the polymers in one cluster show similar properties. Comparing the result of the clustering algorithms with the melt instabilities should, in the best case establish a direct link between melt instability and cluster.

### 2.2. Data Preparation

In this work, the data from the publication by Filipe et al.<sup>[3]</sup> is used, as they investigated a variety of polyethylene samples. They analyzed these samples using large amplitude oscillatory shear (LAOS)<sup>[24]</sup> and Fourier transform rheology.<sup>[25]</sup> Our analysis is only based on Tables 1 and 2 from ref. [3]).

Comparing Table 2 with Table 1 it becomes apparent that the samples LCB 5 and LCB 2 are missing. However, with the help of Figure 3 in the corresponding paper, it was possible to add them to Table 4. The characteristic frequencies obtained from on-line measurements with the Sharkskin option of a capillary rheometer have been compared to the extrudate by investigation under a microscope<sup>[26–29]</sup> and by optical Fourier analysis,<sup>[30]</sup> validating this approach. This proves the ability of the sharkskin option to correctly identify instability with the standard pressure deviation  $\sigma_p$  and the corresponding instability frequency  $f$ . Therefore only columns 3 and 4 from Table 2 will be considered in the following analysis and the LAOS data will not be considered anymore. By including these two features directly linked to the instability we expect samples to be clustered as expected. That is the samples with a rather high instability frequency belong to a cluster which can be associated to sharkskin and samples with a rather low instability frequency would be clustered in another cluster. However, this is only done to get a better understanding of the impact of the data on the clustering. The final aim would be to see if it is possible to obtain two clusters, one that can be associated to sharkskin, the other to stick slip by only using the data from Table 1. To obtain the standard pressure deviation  $\sigma_p$  and the corresponding instability frequency  $f$  Filipe et al. used the Göttfert Rheo-tester 2000 capillary rheometer with the sharkskin option.<sup>[3,26,28,31]</sup> A sketch of the sharkskin option is represented in Figure 2. To obtain the characteristic frequency and standard deviation of pressure of the instability (columns 4 and 5 of Table 2) the authors of ref. [3] did the following steps. The slit die has a length of 30 mm, a width of  $W = 3$  mm and a height of

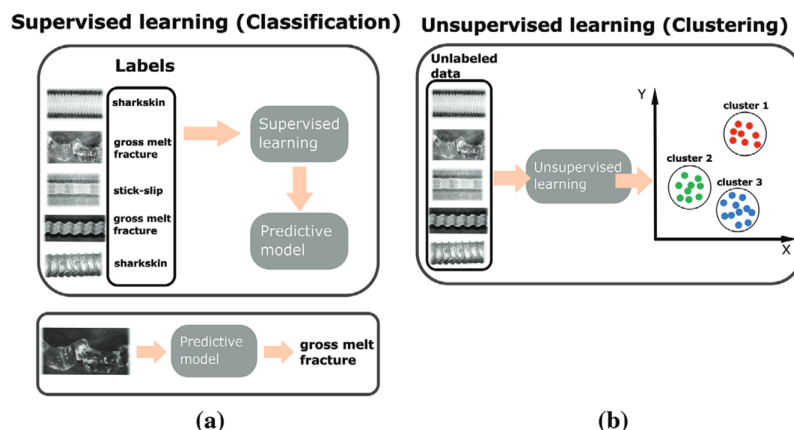


Figure 1. a) Classification (supervised learning) and b) clustering (unsupervised learning).

**Table 1.** Description of samples from ref. [3]. Column 1: Investigated polyethylene samples. The designations L, SCB, LSCB, and LCB denote, linear, short-chain branched, long and short chain branched, and long-chain branched, respectively. Column 2: Weight-average molecular weight  $\overline{M}_w$ . Column 3: Polydispersity  $\overline{M}_w/\overline{M}_n$  as determined by gel permeation chromatography (GPC). Column 6: The total number of all branches (SCB and LCB) of four carbons and longer is identified (measured by melt-state) and represents an upper limit of LCB.

Sample	$\overline{M}_w$ [kg mol <sup>-1</sup> ]	$\frac{\overline{M}_w}{\overline{M}_n}$	Comonomer	Estimated topology	LCB+SCB (> C4)/1000 CH <sub>2</sub>
L1	117	5.1	No	Linear	0.0
SCB2	59	2.1	Yes	SCB	4.5
SCB1	100	2.7	Yes	SCB	2.1
LSCB1	71	2.3	Octene	LCB and SCB	18.9
LCB4	145	8.6	Yes	LCB-low SCB	0.8
LCB3	199	19.0	Very low	LCB-low SCB	0.5
LCB6	206	18.0	Yes	LCB-low SCB	0.9
LCB5	210	20.4	Yes	LCB-low SCB	0.6
LCB2	234	15.6	Very low	LCB	0.5

**Table 2.** Experimental data from ref. [3]. Column 1: Samples. Column 2: Critical strain amplitude  $\gamma_{0,c}$ . Column 3: Maximum non-linear parameter A as determined by LAOS at 180 °C for  $\omega_1/2\pi = \sim 0.1$  Hz. Column 4: Standard deviation of the pressure,  $\sigma_p$ , measured 3 mm after the entry of the slit die (position  $P_1$  [Figure 2]), for capillary measurements performed at a shear rate of 504 s<sup>-1</sup> for  $T = \sim 180$  °C. Column 5: The frequency of the melt flow instability was obtained from the Fourier analysis of the pressure signal at piezoelectric pressure transducer position  $P_1$  (Figure 2), for 180 °C and for an apparent shear rate of 504 s<sup>-1</sup>.

Samples	$\gamma_{0,c}$	A [%]	$\sigma_p$ [bar]	$f_{inst}$ [Hz]	Instability
L1	1.00	7.0	9.65	1.93	Sharkskin
SCB2	>3.00	1.0	2.15	3.08	Sharkskin
SCB1	1.25	4.0	0.85	4.81	Sharkskin
LSCB1	1.50	8.0	1.06	3.08	Sharkskin
LCB4	0.75	10.0	2.02	6.16	Sharkskin
LCB3	0.45	11.0	2.75	0.243	Stick-slip
LCB6	0.50	11.0	11.08	0.075	Stick-slip

$H = \sim 0.3$  mm. For this slit die 3 piezoelectric transducers with a sampling rate of 20 kHz are located along the die. They are lo-

cated 3, 15, and 27 mm from the die entry. At the indicated shear rate of 504 s<sup>-1</sup> at 180 °C all the samples showed a melt instability (sharkskin or stick-slip) with the corresponding standard pressure deviation  $\sigma_p$ . Applying a Fourier transform to the pressure signal of the three pressure transducers along the die leads to the instability frequency  $f$  as indicated in Table 2. As all the three highly sensitive pressure transducers record the melt instability induced pressure fluctuations it is enough to only consider one of the pressure transducers. In this case the pressure transducer closed to the die entry has been selected ( $P_1$  in Figure 2). The pressure signal recorded over time is converted in the frequency space using the Fourier transform. In the case of an instability the frequency spectrum shows a peak at a characteristic frequency that can be associated to an instability.

Concerning the data, another problem is column 5 in Table 1 about the estimated topology. This column needs to be rewritten to be used by the clustering algorithm as they require numbers and not text as input. A rather simple approach is used where column 5 (estimated topology) is split in two columns, namely SCB and LCB.

With respect to Table 3, if no branching is present at all SCB and LCB are both set to 0. If only SCB branching is present in the



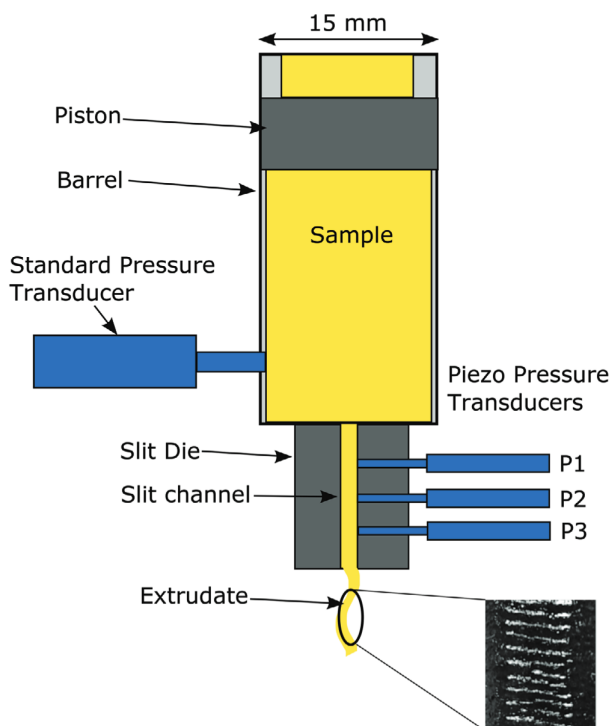


Figure 2. Sharkskin option of a capillary rheometer.

Table 3. Conversion of column 5 (“Estimated topology”) of Table 1 into SCB and LCB.

Estimated topology	SCB	LCB
Linear	0	0
SCB	1	0
SCB	1	0
LCB and SCB	⇒ 0.5	0.5
LCB-low SCB	0.9	0.1
LCB-low SCB	0.9	0.1
LCB-low SCB	0.9	0.1
LCB-low SCB	0.9	0.1
LCB	0	1

SCB column the value is set to 1 and in the corresponding row of the LCB column to 0. As it is not clear what LCB-low means, in this case it was simply set to 0.1 in the LCB column and 0.9 in the SCB column. Here there is definitely room for improvement but the program still should be able to handle these data. The final data table used for the clustering algorithm is Table 4.

It should be noted that the final aim is to cluster all the samples without using columns 7 and 8 in Table 4, as they are direct experimental evidence of a melt instability. However, they are initially considered to verify, via the feature selection procedure, how crucial they really are for the clustering of the samples. In column 9, the sharkskin and stick-slip instabilities have been enumerated to simplify their identification in the graphs in Section 6. In machine learning jargon the columns in Table 4 are referred to as

features, whereas each column represents one feature. For machine learning applications, the data set under investigation is rather small, therefore even computationally expensive clustering algorithms can be used. As unsupervised algorithms are employed it is crucial to keep in mind, that the column “instability” (Table 4) will not be used during the clustering of the algorithm. It will only later be used to check if the clustering was successful. As the instability is available (labeled data), it would be possible to use a supervised approach, namely a classification algorithm. In this case, classification algorithms like support vector machines (SVM) could be used to identify two classes, namely stick slip and sharkskin. After training the supervised algorithm, a new sample could then simply be assigned to one of the two classes. Clustering algorithms however look for similarities on their own and therefore could help to discover additional currently unknown similarities between samples. Instead of choosing between clustering and classification, they could also be combined. In this case the clustering algorithm would group the samples in specific clusters, hence, labeling them, where each sample belonging to a specific cluster has the same label. Then the labeled data are fed into a classification algorithm. This is reasonable for a large number of clusters. As in our case only two clusters are expected we restrict ourselves only to the clustering part.

### 3. Feature Scaling and Feature Selection Procedures

There are many different steps that need to be considered before a specific algorithm can make useful predictions. For more details on machine learning we highly recommend the book Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow from Aurélien Géron.<sup>[11]</sup>

#### 3.1. Feature Scaling

Often features have varying degrees of magnitude, range, and units. The weight-average molecular weight  $\bar{M}_w$  ranges for example takes values ranging from 59–234 kg mol<sup>-1</sup> whereas the frequency of the instability  $f_{inst}$  ranges from 0.08–6.16 Hz. Clustering algorithms (K-means, mean-shift...) calculate the distance between features and if the distance is below a given threshold, a specific sample is assigned to a given cluster. Therefore they are sensitive to features scaling. If features have very different scales the features with the largest values are typically dominating although they are not necessarily the most important ones. Therefore feature scaling methods needs to be applied to solve this problem. The two common feature scaling methods are normalization and standardization.

##### 3.1.1. Normalization

A feature  $X$  is normalized to  $X_{norm}$  according to Equation (1)

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

This shifts and rescales the values to  $X_{norm} \in [0, 1]$



Table 4. Data used for clustering algorithms.

Sample	$\bar{M}_w$ [kg mol <sup>-1</sup> ]	$\frac{\bar{M}_w}{\bar{M}_n}$	SCB	LCB	LCB+SCB (> C4)/1000 CH <sub>2</sub>	$\sigma_p$ [bar]	$f_{inst}$ [Hz]	Instability
L1	117	5.1	0	0	0.0	9.65	1.93	Sharkskin 1
SCB2	59	2.1	1	0	4.5	2.15	3.08	Sharkskin 2
SCB1	100	2.7	1	0	2.1	0.85	4.81	Sharkskin 3
LSCB1	71	2.3	0.5	0.5	18.9	1.06	3.08	Sharkskin 4
LCB4	145	8.6	0.9	0.1	0.8	2.02	6.16	Sharkskin 5
LCB3	199	19.0	0.9	0.1	0.5	2.75	0.243	Stick-slip 1
LCB6	206	18.0	0.9	0.1	0.9	11.08	0.075	Stick-slip 2
LCB5	210	20.4	0.9	0.1	0.6	8.6	0.24	Stick-slip 3
LCB2	234	15.6	0	1	0.5	11	0.08	Stick-slip 4

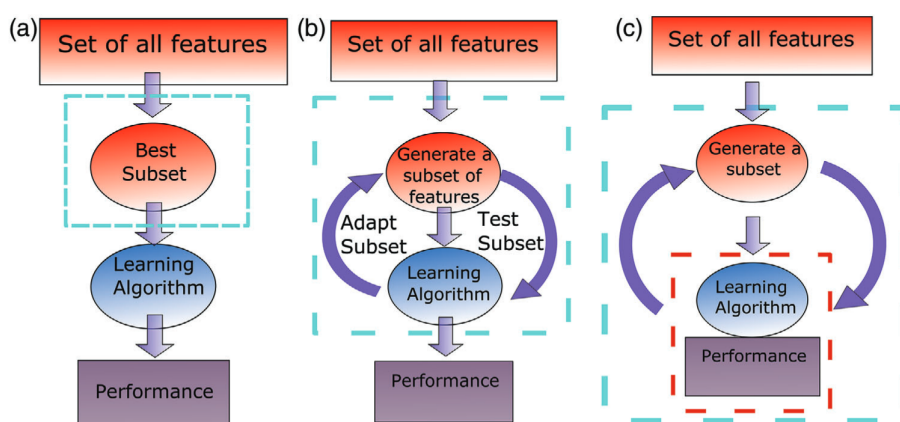


Figure 3. a) Filter method, b) wrapper method, and c) embedded method.

### 3.1.2. Standardization

A feature  $X$  is standardized to  $X_{std}$  according to Equation (2)

$$X_{std} = \frac{X - \mu}{\sigma} \quad (2)$$

$X_{std}$  values are centered around the mean ( $\mu$ ) and the standard deviation ( $\sigma$ ) of the feature values with a unit standard deviation.

## 3.2. Feature Selection

Feature selection refers to the process where only those features are selected (manually or automatically) which contribute most to the correct prediction.<sup>[11,32]</sup> Feature selection has the following advantages:

- 1) Reduction of measurements as some features, in our case properties obtained from experimental results, might have only a little impact on the outcome of the clustering.
- 2) Reduction of overfitting as less redundant data reduces the probability to make decisions based on noise.
- 3) Reduced training time as the number of features, hence the number of data points is reduced
- 4) Hierarchy of most important features.

Typically, feature selection methods are applied before training the algorithm to reduce the number of features due to the before mentioned reasons. As our data set is very small, there is no need to reduce it, therefore feature selection methods are rather used to understand which features have the strongest impact on the clustering. There exist feature selection methods for unsupervised learning algorithms, however most easily accessible machine learning libraries like scikit-learn offers only supervised feature selection methods. This is however no problem as clustering the samples leads to labeled data as all the samples belonging to a specific cluster are labeled accordingly. The initially unsupervised problem has now become a supervised one and hence additional algorithms can be applied. In the case of supervised learning there exist three kinds of feature selection methods. Namely, filter methods, wrapper methods and intrinsic/embedded methods (Figure 3). These three methods differ in the way they select the best features. As at least one of each method will be later applied, a brief explanation of the advantages and differences is given.

### 3.2.1. Filter Method

Filter methods use statistical techniques to evaluate the relationship between each input and target variable. Based on these scores the best features are selected. As can be seen in Figure 3a,

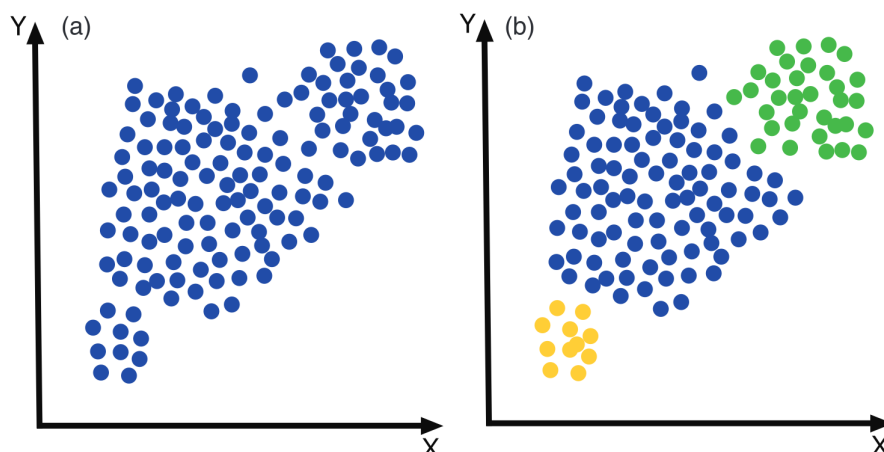


Figure 4. a) Before clustering and b) after clustering.

illustrated by the blue dashed box, the filter method directly selects the best subset of features before passing it to the learning algorithm. The advantage is that filter methods are very fast, but on the other hand, the statistical measures employed are typically calculated for one input variable at a time with respect to the target variable. Therefore interactions between input variables are not considered in the filtering process (univariate statistical measure).

### 3.2.2. Wrapper Method

In contrast to filter methods, wrapper methods need a machine learning algorithm and use its performance as evaluation criteria. There exist several approaches to determine the best subset of features. The most common ones are:

- 1) Forward selection: Start with an empty set of features, then the best feature is selected, next the second best, and so on.
- 2) Backward elimination: Start with all the features and remove the worst one at each step.
- 3) Combination of forward and backward elimination: Add the best and remove the worst feature.
- 4) Recursive feature elimination: Performs a greedy search to find the best performing feature subset.

Wrapper methods usually provide the best subset of feature for a particular model. Unfortunately they are computationally very expensive. For the small data sets in this paper they however can be applied easily.

### 3.2.3. Intrinsic/Embedded Method

Embedded methods are in contrast to the two other feature selection methods integrated part of the algorithm (Figure 3c). This means they do not necessarily train the algorithm on a reduced feature set (except when a filter method is applied prior to the algorithm) but the feature selection is part of the applied algorithm. This is for example the case for decision trees, Random

Forest and regularized models like the Ridge or LASSO Regression.

## 4. Clustering Algorithms

Contrary to classification, clustering is an unsupervised machine learning technique that involves the grouping of data points. Data points that are part of the same group should have similar properties, while data points in different group should have dissimilar properties. This concept is illustrated in Figure 4a,b. In Figure 4a it is apparent that two clusters are present but the third one is not that obvious.

In total nine different clustering algorithms are used, namely: Agglomerative hierarchical clustering,<sup>[33]</sup> density-based spatial clustering of application with noise (DBSCAN), Gaussian mixture, K-means, ordering points to identify the clustering structure (OPTICS), mean shift, SPECTRAL, mini-batch K-means, balanced iterative reducing and clustering using hierarchies (BIRCH). All these algorithms are part of the scikit learn library and will not be discussed in detail here. Nevertheless, for some selected algorithms the advantages and disadvantages will be highlighted.

### 4.1. K-Means Clustering

Probably the most popular algorithm<sup>[34–36]</sup> due to its simplicity and linear time complexity  $\mathcal{O}(n)$ . On the other hand, one of the major drawbacks is that the number of cluster ( $k$ ) needs to be indicated in advance. The typically random initialization of the cluster centers might lead to different results for different runs. It works best for spherical clusters of similar size. In this work, the number of clusters is known in advance. Two clusters are expected, one corresponding to sharkskin and another one to stick slip. Fortunately, there exist some methods for the K-means algorithms to obtain the right number of clusters, using an iterative approach, by running the program several times with an increasing number of cluster for each run.<sup>[11]</sup>

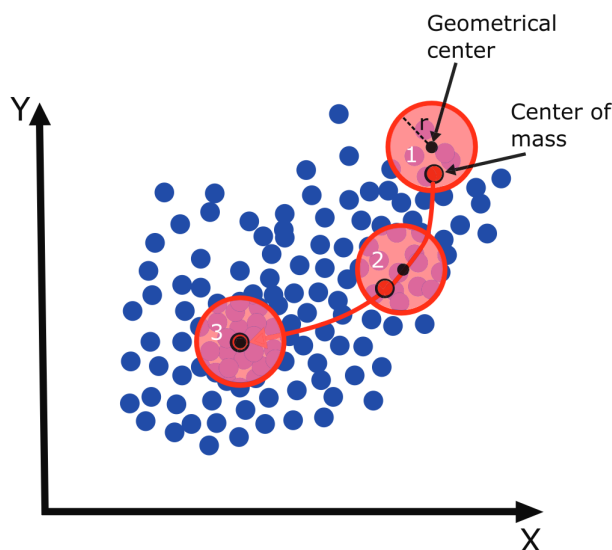


Figure 5. Illustration of the mean-shift clustering algorithm.<sup>[38]</sup>

#### 4.2. Mean-Shift Clustering

The mean-shift algorithm<sup>[37]</sup> is illustrated in Figure 5. The algorithm starts with a circular sliding window of a given radius  $r$  ( $r$  remains constant) centered around a randomly selected starting point (circle 1).

At every iteration, the center of mass of all the data points inside the circle is calculated and the geometrical center is shifted to the center of mass (circle 2). Hence, the circle iteratively shifts to regions of higher density. The sliding window is shifted until there is no direction at which a shift can accommodate more points inside the kernel (circle 3) where geometrical center overlaps with the center of mass. Figure 5 illustrates the case of finding the center of a single cluster. To detect several clusters, the algorithm starts with many sliding windows evenly distributed throughout the whole  $xy$ . This is also one of the advantages of this method as that the number of clusters does not need to be set in advance. On the other hand the performance of the algorithm is crucially dependent on the radius of the sliding window which needs to be set manually. This is indeed the only “human input” that is needed when applying the described technique.

#### 4.3. DBSCAN

DBSCAN<sup>[39]</sup> groups points together that are close to each other (higher density), points with many nearby neighbors. In Figure 6, each color, red, blue, and green corresponds to a cluster. Points in a low density region for example, are considered as outliers and are represented as white in Figure 6. To be considered part of the same cluster, starting from a given point, referred to as core point at least  $n$  other points should be within a distance  $r$  of it where  $r$  is the radius of the neighborhood (circle) and the minimum number of points which form a cluster need to be defined.

#### 4.4. Gaussian Mixture Model

Similar to  $K$ -means, the number of cluster needs to be indicated in a Gaussian mixture model (GMM).<sup>[40]</sup> GMMs assume that the data points are Gaussian distributed (elliptical shape); this is a less restrictive assumption than saying that they are circular as in  $K$ -means. Furthermore, GMMs use probabilities, therefore one data point can belong to multiple clusters with a given probability. A data point could belong with  $X\%$  to cluster 1 and  $Y\%$  to cluster 2.

In Figure 7 each color corresponds to one cluster. The different shadings, darker in the center and lighter at the edges represent the top view of a Gaussian. At the center, in the darker shaded area there is a higher density of points than at the edge.

#### 4.5. Agglomerative Hierarchical Clustering

There exist two sorts of hierarchical clustering, namely agglomerative or divisive clustering. In divisive clustering, also referred to as top-down approach, all observations start as one cluster which is then further divided until each observation corresponds to one cluster. Agglomerative clustering is the exact opposite and is a bottom up approach. In this case each observation starts in its own cluster which is then successively merged into bigger clusters until in the end all the observation form one cluster. The main drawback of this approach is the time complexity of  $O(n^3)$  and memory requirements of  $\Omega(n^2)$ . This makes it typically too slow for even medium data sets. However, as in this work only a very small data set is considered, it can easily be applied. A result of hierarchical clustering is shown in Figure 8a where the molecular weight versus the frequency of the melt instability is plotted. The labels (sharkskin 1, sharkskin 2...) have only been added for visualization and to make sure that the clustering worked correctly. All the violet samples belong to the sharkskin cluster and all the red ones to the stick-slip cluster. The color is the result of the successful clustering algorithm. If, for example, sharkskin 1 would be represented as red dot it means that it would have been assigned to the stick-slip cluster. The corresponding dendrogram to Figure 8a is represented in Figure 8b. The numbers on the horizontal axis corresponds to the sample ID (0–8) where samples 0–4 display the sharkskin instability and samples 5–8 stick-slip. The vertical axis represents the distance between samples or clusters. As example the stick-slip samples from Figure 8a are linked to the dendrogram in Figure 8b. The significant distance between the stick slip and sharkskin cluster becomes obvious.

#### 4.6. Evaluation of Clustering Performance

Evaluating the performance of a supervised learning algorithm is rather simple, as a test set (a subset of the data which has not been used during the training of the models) can be used to test the trained algorithm. As the data are labeled, the outcome of the model can be directly cross checked with the labeled data. Therefore, it can directly be verified how many samples have been classified correctly. For an unsupervised algorithm this is by definition not the case. Clustering algorithms are especially used to



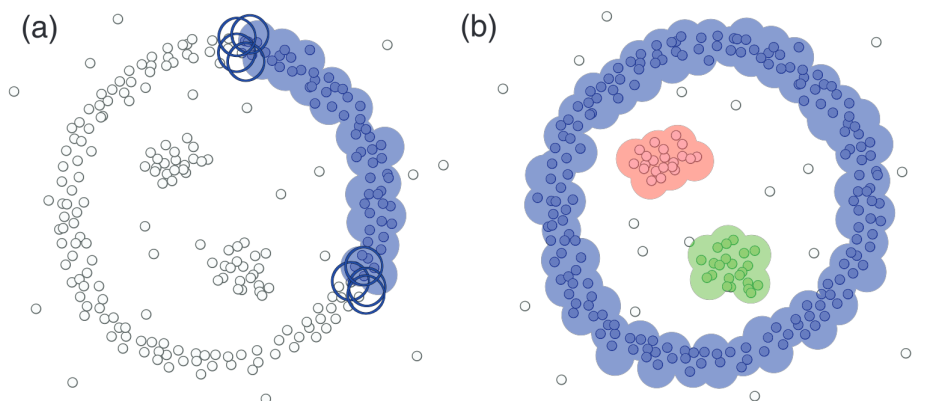


Figure 6. Illustration of cluster detection with DBSCAN in a 2D feature space.

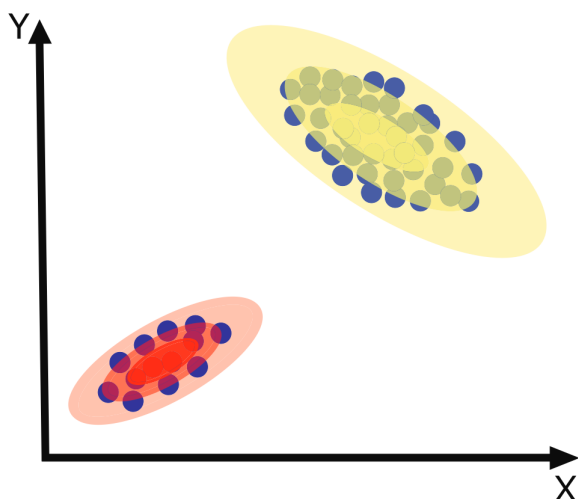


Figure 7. Gaussian mixture model.

discover similarities in data which we were necessarily not aware of. This means that typically a data set is fed to a clustering algorithm and the output would be a list of the data points and their association to the corresponding clusters. However, it is not clear what samples from the same cluster have in common. Therefore, experiments could be done to understand the relation between samples from the same or different clusters. To check the performance in terms of correct clustering several methods could be applied.

- 1) Two features from experimental measurements (standard deviation of pressure and frequency of the instability) from which we know that they directly correlate with the instability are included in the data set. This is done in first step to increase the performance of the clustering algorithms. They will later be removed, to evaluate the performance of the clustering algorithms independent of experimental measurements that are directly linked to the melt instability.
- 2) Although we do not use the labels in the clustering algorithms we can use them to compare the result of the clustering to the

experimental data. In our case we checked if all the samples assigned to a given cluster share the same instability.

- 3) Several clustering algorithms are tested to verify that they lead to a similar output. The result is independent of the real experimental results as only the outputs of the clustering algorithms are compared to each other. This furthermore shows us the strength and weaknesses of the clustering algorithms with respect to a specific data set.
- 4) If the number of clusters is known (as in our case), the number of data points not assigned to a specific cluster can be used to check the performance.

These steps need to be done for each feature scaling method (normalization, standardization) as they might influence the result. It is however crucial that no hyper parameter tuning is performed on the clustering algorithms as otherwise their result would be too specific for a corresponding data set and would not generalize well.

Using the outcome of the experiments in this stage is only done to better understand the outcome of the clustering algorithms and the importance of the features. We create a benchmark which can then be used in the future for other samples.

#### 4.7. Summary of Procedure

A quick summary of the different steps that are required to apply clustering algorithms and feature selection.

- 1) Transform the available data into a reasonable format.
- 2) Apply normalization and standardization and compare the results as described in Section 4.6.
- 3) Use a variety of clustering algorithm to compare their results.
- 4) When all the samples have been assigned to a specific cluster, the cluster corresponds to a label. Meaning that all the samples in the same cluster have the same label. Hence our initially unlabeled data set is now labeled and can be treated as a supervised problem.
- 5) Use different feature selection methods to identify the most important features.

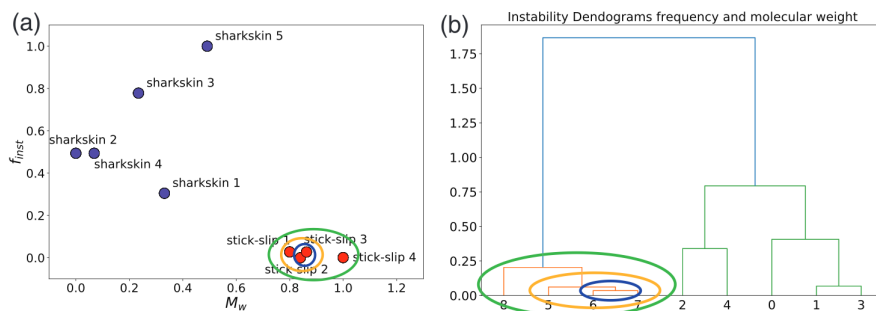


Figure 8. Agglomerative hierarchical clustering. a) Standardized molecular weight versus instability frequency and b) dendrogram corresponding to (a).

- 6) Remove the two features: “standard deviation of pressure” and “frequency of instability” as they are directly linked to the instability and run the clustering algorithms again.

## 5. Algorithm Validation

To create a benchmark, the results of the clustering algorithms are compared to the experimental data. As it is experimentally known that all the samples show an instability it is expected that the number of outliers is zero. In the best case all the samples showing the sharkskin instability should be associated to one cluster and all the samples showing the stick-slip instability to the second cluster. Such a result would look like: [000001111] which corresponds to the last column in Table 4. This means that the first five samples belong to cluster 0 and the last four samples belong to cluster 1. For all the clustering algorithms, some manual hyperparameter tuning has been performed to get the best possible results that is, to reduce the number of outliers. As generally the number of clusters is not known, the aim was to obtain the lowest number of outliers with the least possible amount of fine tuning, as otherwise the parameters would be too specific for general applications.

### 5.1. Raw Data

For the raw data, the results slightly varied for mini-batch K-means and Gaussian mixture which is probably due to the fact that the initial center for the cluster is randomly placed. The result looks like [000001111] (correct) or [000011111]. In the last case, the fifth sample showing sharkskin has been incorrectly assigned to the stick-slip cluster. Spectral, on the other hand, was never able to correctly cluster all the samples.

### 5.2. Standardized Data

For the standardized data, the output for DBSCAN looks like [ - 100 - 10111 - 1]. This means it was able to identify two clusters (cluster 0 and cluster 1), however it could not assign three samples to a specific cluster (samples with -1). Mean shift [000301112] basically created four clusters, where the fourth and the ninth samples are clusters on their own. Also OPTICS [000 - 10111 - 1] could not assign sample 4 and 9 to a specific clusters. These are the same that also Mean shift could not cluster correctly.

Table 5. Effect of data scaling. The numbers indicate how many data points are correctly assigned to the labels.

	Raw	Standardized	Normalized
Agglomerative hierarchical	9	9	9
K-means	9	9	9
Mini-batch K-means	8-9	9	9
DBSCAN	9	6	6
BIRCH	9	9	9
Mean shift	9	7	8
OPTICS	9	7	9
Spectral	6	9	9
Gaussian mixture	8-9	9	9

### 5.3. Normalized Data

For the normalized data, DBSCAN was able to identify two clusters, but it could not assign three samples to a specific cluster. As for the standardized data, the same three samples [ - 100 - 10111 - 1] could not be assigned to a specific cluster. For mean shift, the result looks like [000001112], very similar to the result for the standardized data set, but in this case the fourth sample has been correctly clustered, but a third cluster has been created for the ninth sample.

There is not a big difference between the clustering of standardized, normalized and raw data. Normalization slightly performs better as only two out of nine algorithms could not cluster all the samples compared to three out of nine for raw and standardized data. Therefore normalization became the method of choice. The impact that the scaling has on the result is clearly visible in Table 5. The fact that the clustering results based on raw data is similar to the normalized and standardized is only a lucky coincidence for this data set as will be shown later. For the normalized data, only DBSCAN and Mean shift fail to assign all the samples to the corresponding clusters. This means that the choices of the clustering algorithm is not very crucial and even the most basic clustering algorithms as K-means can be used in this case.

## 6. Feature Selection Results and Discussion

In this project, the samples showing instabilities are known (last column in Table 4). This is typically not the case, actually it is

Table 6. Ranking of all the feature selection methods.

SelectKBest f_classif (Filter)		SelectKBest chi-square (Filter)	
Ranking	Score	Ranking	Score
$\frac{\overline{M}_w}{\overline{M}_n}$	72.2	$\frac{\overline{M}_w}{\overline{M}_n}$	2.9
$\overline{M}_w$	36.6	$\sigma_p$	2.7
$f$	18.6	$f$	2.3
$\sigma_p$	16	$\overline{M}_w$	1.8
LCB+SCB (> C4)/1000 CH <sub>2</sub>	1.4	LCB+SCB (> C4)/1000 CH <sub>2</sub>	0.8
LCB	0.8	LCB	0.4
SCB	0.0003	SCB	0.00008
RFE Logistic Regression (Wrapper)		RFE linear SVC (Wrapper)	
Ranking	Score	Ranking	Score
$\frac{\overline{M}_w}{\overline{M}_n}$		$\frac{\overline{M}_w}{\overline{M}_n}$	
$\sigma_p$		$f$	
$\overline{M}_w$		$\overline{M}_w$	
$f$		$\sigma_p$	
LCB+SCB (> C4)/1000 CH <sub>2</sub>		LCB	
LCB		SCB	
SCB		LCB+SCB (> C4)/1000 CH <sub>2</sub>	
RFE Decision Tree (Wrapper)		Ridge Regression (Embedded)	
Ranking	Score	Ranking	Score
$f$		$\frac{\overline{M}_w}{\overline{M}_n}$	0.34
$\sigma_p$		$f$	0.30
LCB		$\overline{M}_w$	0.26
SCB		$\sigma_p$	0.23
LCB+SCB (> C4)/1000 CH <sub>2</sub>		LCB	0.08
$\overline{M}_w$		SCB	0.08
$\frac{\overline{M}_w}{\overline{M}_n}$		LCB+SCB (> C4)/1000 CH <sub>2</sub>	0.05
$\overline{M}_n$		ExtRaT Classifier 1000 est. (Embedded)	
ExtRaT Classifier 100 est. (Embedded)		Ranking	Score
Ranking	Score	$\frac{\overline{M}_w}{\overline{M}_n}$	0.3
$\overline{M}_w$	0.28	$f$	0.22
$\frac{\overline{M}_w}{\overline{M}_n}$	0.27	$\overline{M}_w$	0.21
$f$	0.22	$\sigma_p$	0.18
$\sigma_p$	0.15	LCB+SCB (> C4)/1000 CH <sub>2</sub>	0.04
LCB	0.05	LCB	0.04
SCB	0.02	SCB	0.01
LCB+SCB (> C4)/1000 CH <sub>2</sub>	0.01		

RFE, recursive feature elimination; SVC, support vector classification; ExtRaT, extremely randomized trees classifier; est, estimators.

by definition never the case for an unsupervised problem. Applying a clustering algorithm to the data, leads to two clusters. Comparing the clusters with experimental results confirmed the link between a specific cluster and a given melt instability. (e.g., [000001111] (correct)) permits the use of supervised learning methods, like feature selection. As the unsupervised clustering problem (Table 4 without column “instability”) becomes now a supervised classification problem (Table 4 with the column “instability”). Hence supervising feature selection methods can be applied. There also exist unsupervised feature selection methods, they are however less common and typically not yet implement in common libraries. The employed feature selection methods can

be divided in three groups as explained in Section 3.2, namely filter, wrapper, and embedded methods. The results will show that every method leads to slightly different results. Therefore, to obtain a final ranking, to each of the seven features a value is assigned (Table 6). For a specific method, the first ranked feature obtains seven points, the second ranked six, and so on. In the end the scores of all the features for every feature selection method are averaged and a final ranking is obtained. Some feature selection methods even assign a score to each feature. If this score is available it will also be shown. This allows to better understand how different features compare to each other. It should however be noted that the score from one feature selection method



**Table 7.** Final ranking after averaging the results of all the feature selection methods.

	Final ranking	Average score
1	$\frac{\bar{M}_w}{\bar{M}_n}$	6.2/7
2	$f_{inst}$	5.5/7
3	$\bar{M}_w$	4.9/7
4	$\sigma_p$	4.7/7
5	LCB	2.8
6	LCB + SCB(> C4)/1000 CH <sub>2</sub>	1.8
7	SCB	1.8

**Table 8.** Clustering without using instability frequency and pressure data. The numbers indicate how many data points are correctly assigned to the labels.

	Normalized
Agglomerative hierarchical	9
K-means	9
Mini-batch K-means	9
DBSCAN	0
BIRCH	9
Mean shift	4
OPTICS	6
Spectral	9
Gaussian mixture	7

cannot be compared with the score of another feature selection method.

The final ranking with the averaged score of all the feature selection methods is shown in Table 7.

As can be seen in Table 7, the polydispersity is the most powerful criterion to create distinct clusters, this means it has also the biggest impact on the melt instability. Surprisingly it is even higher ranked than the instability frequency which is a direct experimental measurement of the melt instability. In third and fourth position the molecular weight appears followed by standard deviation of pressure which can also be directly linked to a specific melt instability. Our result is confirmed by Filipe et al.:<sup>[3]</sup> “When considering only molecular weight and polydispersity, and for these particular shear rate, temperature and die geometry, one can state that materials with high-molecular weights and broad PDIs are more predisposed to develop stick-slip instabilities.... Moreover, samples having lower  $\bar{M}_w$  presented shark-skin...” Branching only seems to have a minor effect on the melt instabilities.

As final test, the experimental data from the sharkskin option, namely the instability frequency  $f$  and the standard deviation of pressure  $\sigma_p$  are removed from the data and the unsupervised clustering algorithms are executed again. The question is, if the clustering is still successful after removing two powerful features. The result is shown in Table 8. This time only the performance on a normalized feature set is represented.

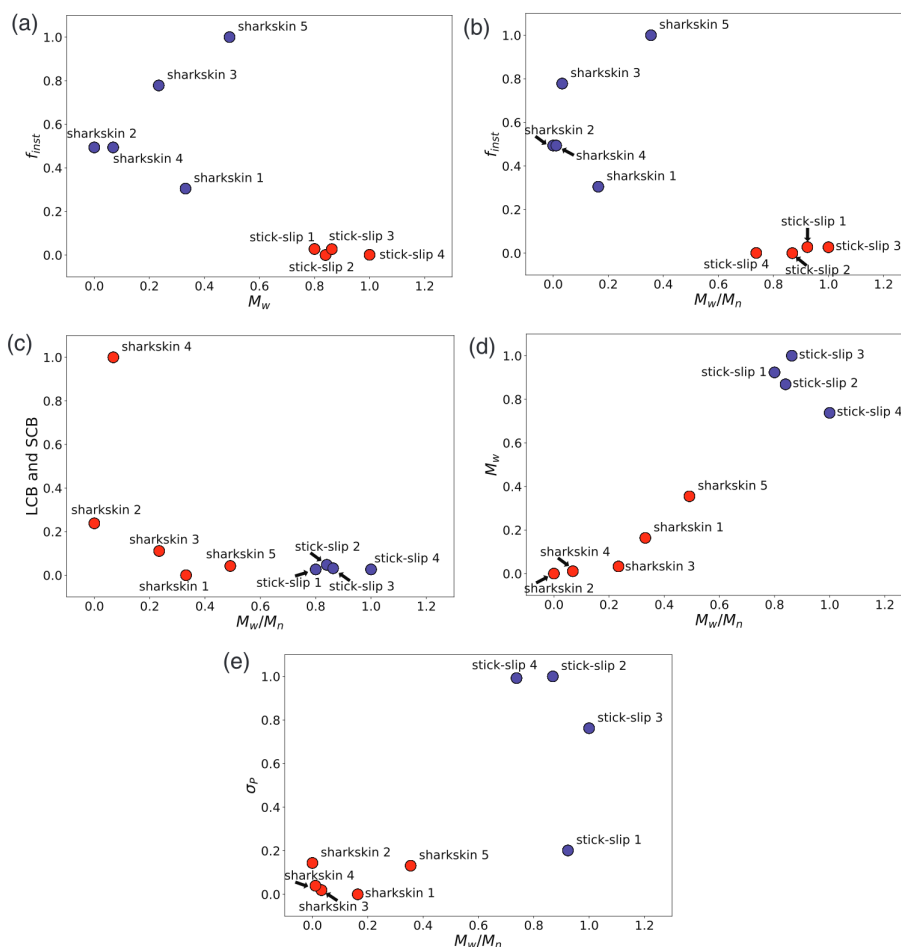
DBSCAN identified every sample to belong to the same cluster. Optics could not assign three samples at all to any of the two clus-

ters and Gaussian mixture assigned two samples to the wrong cluster. In the end five out of nine algorithms were able to correctly cluster all the samples from the reduced data set. In this case the “simpler” clustering algorithms like K-means were able to correctly assign all the samples, as they are less sensitive to outliers compared to DBSCAN for example. For much larger data set with more outliers the results might be different. This approach clearly shows that even with a rather small amount of samples and features very good predictions can be made about the behavior of the polymers during extrusion.

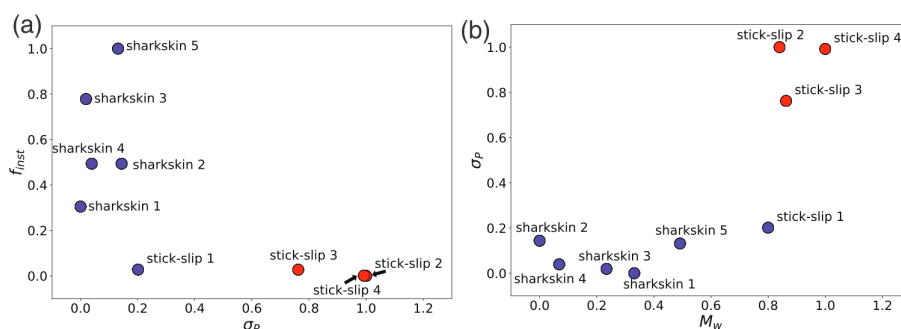
To visualize the results of the feature selection algorithm, some features are plotted versus each other. These figures are the result of the hierarchical agglomerative clustering algorithm. It should be noted that these results depend on the clustering algorithm. This algorithm was selected as it was able to identify all the samples correctly in every case. Compared to the results before, where the clustering algorithm was applied to all the features simultaneously, here it is only applied to two features, as shown in the following Figure 9e. In Figure 9a variety of features are plotted against each other. The colors correspond to the result of the clustering algorithm. All the samples with the same color belong to the same cluster. The labels “sharkskin 1, sharkskin 2...” correspond to the last column in Table 4. They are not used in the clustering algorithm but only to verify if the clustering algorithm assigned the samples to the correct cluster. Typically these labels would not be available for unsupervised problems. But as they exist in this case, they are used to cross check the result. For the feature combinations presented in Figure 9 all the samples are correctly assigned to the corresponding clusters. In contrast to the feature comparison in Figure 9, Figure 10 shows two feature combinations where the clustering algorithm partially failed as the sample labeled stick-slip 1 has been assigned to the shark-skin cluster. However, referring to Tables 5 and 8 it can be seen that the agglomerative clustering algorithm assigned all the samples to the correct cluster if all the features are considered. This is precisely the strength of the unsupervised algorithms, as they compare all the features with each other, and not only two, to establish a feature ranking (Table 7). Therefore, although for some specific feature combinations the clustering is not correct (Figure 10), by taking into account all the features the samples can be correctly clustered. As can be seen in Table 7, the polydispersity  $\bar{M}_w/\bar{M}_n$ , characteristic frequency of the instability  $f_{inst}$ , molecular weight  $\bar{M}_w$  and standard deviation of pressure  $\sigma_p$  are the highest ranked features. As the molecular weight and polydispersity also have rather high numerical values they have a considerable impact on the clustering algorithm if the data are not normalized or standardized. By coincidence, these features are also among the highest ranked features, which explain why the clustering algorithms performed very well on the raw data sets (5). A final, crucial remark is that the samples have all been extruded at 180 °C and for an apparent shear rate of 504 s<sup>-1</sup> (Section 2.2). As can be seen in Table 4, this has not been taken into account in the clustering algorithms proving that the instabilities are an intrinsic property of the polymer.

## 7. Impact of Data Set Size on the Results

With respect to machine learning applications the considered data set is very small. However, it contains a wide variety of sam-



**Figure 9.** Illustration of the clustering results for selected features. The label corresponds to the last column in Table 4. The colors correspond to the result of the clustering algorithm. a) Molecular weight  $M_w$  versus frequency of instability  $f$ ; b) polydispersity  $M_w/M_n$  versus frequency of instability  $f$ ; d) polydispersity  $M_w/M_n$  versus LCB and SCB branching; e) polydispersity  $M_w/M_n$  versus molecular weight  $M_w$ ; and f) polydispersity  $M_w/M_n$  versus standard deviation of pressure  $\sigma_p$ .



**Figure 10.** Illustration of the clustering results for selected features. The label corresponds to the last column in Table 4. The colors correspond to the result of the clustering algorithm. a) Standard deviation of pressure  $\sigma_p$  versus frequency of the instability  $f$  and b) molecular weight  $M_w$  versus standard deviation of pressure  $\sigma_p$ .

ples with different molecular weights, branching, polydispersity, and other properties. As an unsupervised method is employed in this work, there is no training of the algorithms involved compared to supervised algorithms. These algorithms would require large data sets as the data need to be split in 80% training and 20% test data. For a data set it might happen that single data points (samples) would be identified as small clusters or, more commonly, are rejected as they are incorrectly considered as outliers. This can happen independently of the size of the clusters. However, if a few data points are not considered on a small data set it has a bigger impact on the result compared to a large data set. To reduce the probability that a data point is considered as an outlier a variety of different clustering algorithms, ranging from rather basic ones like *K*-means to more sophisticated algorithms as DBSCAN, are tested and compared to each other.

The results in Table 8 show that still five algorithms were able to cluster all data points. This shows that even for such small data sets, our approach works. The results also show that more advanced clustering algorithms are typically more sensitive to outliers, which can be very useful for large data sets but is a drawback for smaller ones. It should furthermore be noted that is not clear from the beginning which algorithms lead to the best results. This is the same in supervised learning applications. In that case, first, a variety of supervised learning algorithms are compared to each other by fitting test data. The algorithms can range from Ridge regression to regression trees over Random Forests to Neural networks. The best algorithm is then selected and is further fine tuned using hyper parameter optimization. The performance of unsupervised learning algorithms cannot be evaluated on labeled data sets. Therefore, we run different algorithms and consider those with the lowest amount of outliers to be the most suitable for our data. It is important to highlight that the aim is not to add data points to a specific cluster but to only reduce the amount of outliers as much as possible. In this work we have the chance that the samples have been labeled through experiments, so we can match later a cluster to a specific instability.

The presented approach and algorithms would not need to be adapted even if the amount of samples is increased tenfold as it would still be a rather small data set for ML applications. The bigger the data set and the wider the range of samples, the more likely additional unknown clusters could be identified. For example, as we expected two instabilities (sharkskin and stick slip), we expected two clusters. For some algorithms, the expected number of clusters even needs to be indicated. However, maybe there exist some other unknown, less pronounced, instabilities which are currently unknown to us, but which could appear as a third or fourth cluster in a larger data set whereas they would be considered as an outlier in a smaller data set. Currently, no further melt instabilities are known for the considered class of materials, but it cannot be excluded that new kind of instabilities appear for other polymers or composites. However, this has no impact on the approach presented in this publication. Only an additional cluster would appear in the results and as no such instabilities have been observed they probably do not exist or are very similar to the already known instabilities and therefore would form a cluster very close to an existing one. The more data points (samples) are clustered the higher the resolution and it would be possible to distinguish clusters that are very close to each other. Finally, we would like to highlight that the data set has not been selected

assuming that the unsupervised learning algorithms could identify exactly two melt instabilities but because of the wide variety of sample parameters.

## 8. Conclusion

In this publication, unsupervised machine learning methods, namely clustering algorithms have been applied to relatively small data sets of samples displaying melt instabilities. The aim was to investigate, based on the topology of polyethylene samples, if a relation between a specific cluster and a melt instability occurring during an extrusion process exists. The main features under investigation were the molecular weight, polydispersity, and branching. In total, nine different clustering algorithms have been applied and depending on the number of features under investigation five to seven clustering algorithms were able to assign all the samples under investigation to a specific cluster. Experimental investigation of the sample showed a clear relation between a specific cluster and a specific melt instability. For this rather small set of features, especially the basic algorithms like *K*-means or BIRCH lead to excellent results. Clustering algorithms sensitive to outliers like DBSCAN performed worse. Clustering the samples, results in a labeled data set as each sample is now assigned to a specific cluster. Even without experimental validation clustering the data leads to a labeled data set, for example, samples belong to cluster 1, other samples to cluster 2. Due to this additional information supervised feature selection methods can be applied, leading to a ranking of the features. The top ranked features have the highest impact on the resulting melt instability. For these samples, the ranking is : 1) polydispersity; 2) molecular weight; and 3) branching. Furthermore, the clustering algorithms did not take the temperature or extrusion shear rate into account, but still were able to cluster the samples in agreement with the melt instabilities. This shows that the melt instability itself is mainly governed by the topology. External factors like temperature and shear rate are also known to have an impact on the melt instabilities, as for example, higher temperatures shift their onset to higher shear rates. However, our results demonstrate that the polymer topology is a crucial component in order to distinguish melt instabilities. Further investigation would be necessary to investigate the effect of shear rates and temperature in detail. All the algorithms and methods employed are a part of the scikit learn library, an open source machine learning library for Python.

## Acknowledgements

This work was funded by the Luxembourg National Research Fund (FNR) and by Goodyear Innovation Center Luxembourg with grant 14263566/EMDD in FNR's BRIDGES program.

## Conflict of Interest

The authors declare no conflict of interest.

## Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.



## Keywords

extrusion, feature ranking, melt instabilities, unsupervised machine learning

Received: October 28, 2022

Revised: January 31, 2023

Published online:

- [1] J.-F. Agassant, D. R. Arda, C. Combeaud, A. Merten, H. Muenstedt, M. R. Mackley, L. Robert, B. Vergnes, *Int. Polym. Process.* **2006**, 21, 239.
- [2] B. Vergnes, *Int. Polym. Process.* **2015**, 30, 3.
- [3] S. Filipe, I. Victorias, M. Wilhelm, *Macromol. Mater. Eng.* **2008**, 293, 57.
- [4] D. E. Rumelhart, J. L. McClelland, in *Learning Internal Representations by Error Propagation*, MIT press, Cambridge, MA **1987**, pp. 318–362.
- [5] G. V. Cybenko, *Math. Control, Signals Syst.* **1989**, 2, 303.
- [6] H. J. Kelley, *ARS J.* **1960**, 30, 947.
- [7] A. E. Bryson, in *Proc. Harvard University Symp. on Digital Computers and Their Applications*, vol. 72, **1961**, p. 22.
- [8] S. Dreyfus, *J. Math. Anal. Appl.* **1962**, 5, 30.
- [9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, et al., *Proc. of the 12th USENIX Conf. on Operating Systems Design and Implementation*, USENIX Association, USA **2015**, pp. 265–283. <https://www.tensorflow.org/>, Software available from tensorflow.org.
- [10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, *J. Mach. Learn. Res.* **2011**, 12, 2825.
- [11] A. Géron, *Hand-On Machine Learning with Scikit-Learn, Keras, and Tensorflow*, 2nd ed., O'Reilly Media, Inc, Sebastopol, CA **2019**.
- [12] B. Sahoo, S. De, B. Meikap, *Int. J. Min. Sci. Technol.* **2017**, 27, 379.
- [13] A. R. Shaik, W. AlAmeri, A. AlSumaiti, M. Muhammad, N. C. Thomas, presented at *Abu Dhabi Int. Petroleum Exhibition and Conf.*, Society of Petroleum Engineers, Abu Dhabi, UAE, November **2019**. <https://doi.org/10.2118/197166-ms>.
- [14] A. S. Alqahtani, *Master's Thesis*, The University of Texas at Austin **2019**.
- [15] Y. Li, S. Tang, B. C. Abberton, M. Kröger, C. Burkhart, B. Jiang, G. J. Papakonstantopoulos, M. Poldneff, W. K. Liu, *Polymer* **2012**, 53, 5935.
- [16] Z. Tariq, M. Murtaza, M. Mahmoud, *ACS Omega* **2020**, 5, 17646.
- [17] Y. Saad, D. Gao, T. Ngo, S. Bobbitt, J. R. Chelikowsky, W. Andreoni, *Phys. Rev. B* **2012**, 85, 104104.
- [18] M. E. Abbassi, J. Overbeck, O. Braun, M. Calame, H. S. J. van der Zant, M. L. Perrin, *Commun. Phys.* **2021**, 4, 50.
- [19] C. Lopez, S. Tucker, T. Salameh, C. Tucker, *J. Biomed. Inf.* **2018**, 85, 30.
- [20] M. Mjahed, *Nucl. Instrum. Methods Phys. Res., Sect. A* **2006**, 559, 199.
- [21] Q. Wei, R. G. Melko, J. Z. Y. Chen, *Phys. Rev. E* **2017**, 95, 032504.
- [22] Q. Parker, D. Perera, Y. W. Li, T. Vogel, *Phys. Rev. E* **2022**, 105, 035304.
- [23] D. Bhattacharya, T. K. Patra, *Macromolecules* **2021**, 54, 3065.
- [24] R. Ewoldt, A. Hosoi, G. McKinley, *Annu. Trans. Nord. Rheol. Soc.* **2007**, 15, 3.
- [25] M. Wilhelm, *Macromol. Mater. Eng.* **2002**, 287.
- [26] H. Palza, I. F. Naue, S. Filipe, A. Becker, J. Sunder, A. Göttfert, M. Wilhelm, *Kautsch. Gummi Kunstst.* **2010**, 63, 456.
- [27] H. Palza, S. Filipe, I. F. Naue, M. Wilhelm, *Polymer* **2010**, 51, 522.
- [28] I. F. C. Naue, R. Kádár, M. Wilhelm, *Macromol. Mater. Eng.* **2015**, 300, 1141.
- [29] H. Palza, I. F. C. Naue, M. Wilhelm, *Macromol. Rapid Commun.* **2009**, 30, 1799.
- [30] A. Gansen, M. Rehor, C. Sill, P. Polinska, J. Dheur, J. Hale, J. Baller, *J. Appl. Polym. Sci.* **2020**, 137, 48806.
- [31] I. F. Naue, *Ph.D. Thesis*, Institut für Technische Chemie und Polymerchemie Karlsruhe, Karlsruhe, Baden-Württemberg **2013**.
- [32] Datacamp, [www.datacamp.com/community/tutorials/feature-selection-python](https://www.datacamp.com/community/tutorials/feature-selection-python) (accessed: October 2021).
- [33] F. Nielsen, in *Introduction to HPC with MPI for Data Science, Undergraduate Topics in Computer Science*, Springer International Publishing, Berlin **2016**, pp. 195–211.
- [34] S. Lloyd, *Bell Telephone Laboratories Paper* **1957**.
- [35] J. MacQueen, *IEEE Transactions Information Theory* **1982**, 28, 129.
- [36] H. Steinhaus, *Bull. Acad. Pol. Sci.* **1956**, 4, 801.
- [37] Y. Cheng, *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, 17, 790.
- [38] N. S. Chauhan, *Machine Learning*, KDnuggets, [www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html](https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html) (accessed: March 2021).
- [39] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, in *Proc. of the Second Int. Conf. on Knowledge Discovery and Data Mining (KDD-96)*, AAAI Press, Palo Alto, CA **1996**, pp. 226–231.
- [40] D. P. G. J. McLachlan, *Finite Mixture Models*, John Wiley & Sons, Hoboken, NJ **2000**.