Research Article

Rodrigo Q. Albuquerque, Christian Brütting, Tobias Standau, and Holger Ruckdäschel*

A machine learning investigation of low-density polylactide batch foams

https://doi.org/10.1515/epoly-2022-0031
received December 03, 2021; accepted January 28, 2022

Abstract: Developing novel foams with tailored properties is a challenge. If properly addressed, efficient screening can potentially accelerate material discovery and reduce material waste, improving sustainability and efficiency in the development phase. In this work, we address this problem using a hybrid experimental and theoretical approach. Machine learning (ML) models were trained to predict the density of polylactide (PLA) foams based on their processing parameters. The final ML ensemble model was a linear combination of gradient boosting, random forest, kernel ridge, and support vector regression models. Comparison of the actual and predicted densities of PLA systems resulted in a mean absolute error of 30 kg·m^{-3} and a coefficient of determination (R^2) of 0.94. The final ensemble model was then used to explore the ranges of predicted density in the space of processing parameters (temperature, pressure, and time) and to suggest some parameter sets that could lead to low-density PLA foams. The new PLA foams were produced and showed experimental densities in the range of 36–48 kg·m^{-3}, which agreed well with the corresponding predicted values, which ranged between 38 and 54 kg·m^{-3}. The experimental–theoretical procedure described here could be applied to other materials and pave the way to more sustainable and efficient foam development processes.

Keywords: polylactide foams, biopolymers, sustainability, machine learning, model prediction

1 Introduction

The efficient search for new polymeric materials with desired properties is the key to finding sustainable solutions to the increasingly complex problems of modern society. The trial-and-error method of finding better materials is not environmentally friendly because it leads to much waste of materials until good candidates are finally found. Moreover, this inefficient method consumes important human resources and could delay a market introduction. Molecular dynamics or multiscale methods can be used to develop new polymeric materials *in silico* without experimental data (1). This avoids material waste and can potentially accelerate material discovery. However, screening many rather complex polymer systems with *in silico* methods is nontrivial and still not very accurate and also time consuming. Finding the best balance between experimental and theoretical work to develop new materials with tuned properties is still a scientific challenge today.

One way to address this challenge is supervised machine learning (ML). Available experimental data can be used to build databases, which in turn can be used to train ML models that use processing variables (e.g., temperature and pressure) to predict the final property of interest. The trained ML model then suggests new processing variables that should be used in the next experiment, hopefully resulting in a material that has the desired property. In practice, the results of each new ML suggestion are used continuously to further train the ML model and make the predictions even more accurate. Continuously training ML models and using their predictions in the manufacturing process of novel materials in a fully autonomous way represents another step toward digitalization that will soon change the way materials are discovered and manufactured. The great potential of digitalization for materials science and polymers in general has been well summarized in the work of Kimmig et al. (2).

Polylactide (PLA) is often referred to as the most promising biobased and biodegradable polymer because it has similar properties to polystyrene (3) and has a rather

* **Corresponding author: Holger Ruckdäschel,** Department of Polymer Engineering, University of Bayreuth, Universitätsstraße 30, 95447 Bayreuth, Germany; Neue Materialien Bayreuth GmbH, Gottlieb-Keim-Straße 60, 95448 Bayreuth, Germany, e-mail: holger.ruckdaeschel@uni-bayreuth.de
Rodrigo Q. Albuquerque: Department of Polymer Engineering, University of Bayreuth, Universitätsstraße 30, 95447 Bayreuth, Germany; Neue Materialien Bayreuth GmbH, Gottlieb-Keim-Straße 60, 95448 Bayreuth, Germany
Christian Brütting, Tobias Standau: Department of Polymer Engineering, University of Bayreuth, Universitätsstraße 30, 95447 Bayreuth, Germany

low carbon footprint compared to other polymers (4). Usually, the polymer consists of a mixture of the two enantiomers L-lactide and D-lactide. In commercial grades L-PLA usually predominates (5). With increasing D-content, the glass transition temperature T_g, the melting temperature T_m, and the crystallinity decrease, as this leads to crystal breakage and consequently to a higher amorphous fraction (6–9). High D-contents of 10–12% (5) (and above) lead to fully amorphous PLA. Conversely, low D-contents would lead to higher crystallinities and crystallization rates (10). Litauszki et al. (11) have shown that the heat deflection temperature can be increased by higher crystallization in PLA grades with lower D-content. Other properties are also influenced by the D-content such as transparency or mechanical strength.

In the last decades, intensive research activities, especially in the field of PLA foaming (12), have been recorded. In academia, batch foaming experiments are often performed using autoclaves. A distinction can be made between temperature-induced and pressure-induced foaming. In both cases, foaming is induced by subjecting a gas-loaded sample to thermodynamic instability. In temperature-induced batch foaming, the polymer sample is saturated at high pressure and then transferred to a hot medium (e.g., oil or glycerol, which is usually heated above the T_g of the polymer), where expansion takes place. In the pressure-induced method, the sample is placed in a high-pressure vessel at elevated temperatures and expansion occurs by rapid depressurization. These and other foaming methods are described in more detail in various review articles (12,13).

Chemical modifications are often used to foam PLA in order to increase the melt strength and thus the expandability (14–16). However, with the right thermal and rheological behavior, neat PLA can be foamed very well, as shown in a previous study comparing several commercial PLA grades in autoclave foaming (pressure-induced) experiments (17). It is worth noting that PLA with a high D-content (i.e., 12%) can be foamed at much lower saturation temperatures (<100°C) than grades with lower D-content but the same molecular weight. With autoclave tests, it is possible – even with a small amount of material – to perform numerous experiments. For each material in the earlier study mentioned (17), there are datasets that are constantly being expanded. However, if one has sufficient data, one can apply ML techniques to predict optimal processing parameters that will result in foams with the desired target properties. Mapping low-density foam regions with respect to the appropriate processing parameters using ML modeling would enable the development of novel low-density PLA foams with a minimum of new experimentation.

Two ML models (K-nearest neighbors, KNN, and artificial neural network) have recently been used to predict the Biot parameters of porous absorbers made of PLA (18). Mulrennan et al. (19) reported the use of another ML model using random forest (RF) and principal component analysis (PCA) to predict the yield stress of extruded PLA sheets and suggested the development of a quality assurance tool to determine if a product is out of specification. While some ML models can be found in the literature that have been used for various PLA materials, the application of ML models to support the design of PLA foams with a specific target density is, to our knowledge, unexplored.

The aim of this work is to perform a joint experimental and theoretical investigation to understand and model the dependence of the density of PLA foams on their processing parameters. We show here that single ML models such as RF and gradient boosting (GBR) can predict the density of PLA foams with a small error. We combine these models via an ensemble approach (*vide infra*) to map low-density regions of PLA foams as a function of temperature, pressure, and time. Finally, we discuss how the ML ensemble model can be used to produce new low-density PLA foams.

2 Materials and methods

2.1 Samples and their properties

In this study, nine different grades of PLA from NatureWorks LLC (Minnetonka, MN, USA) were used and are summarized in Table 1. The sample designation is a combination of the abbreviated putative use and the D-content, as known from a previous study of ours in which the main properties were determined (17). The specimens (10 mm × 20 mm × 1 mm) were prepared by hot-pressing.

2.2 Autoclave foaming

Autoclave experiments were performed in an electrically heated, custom-built vessel with digital temperature and pressure couplings. The melt-pressed specimens were placed in a sample tray and then placed in an autoclave, which was tightly sealed. Supercritical CO_2 was used as the blowing agent. A dual arrangement of Teledyne ISCO 260D syringe pumps (Thousand Oaks, CA, USA) was used to apply a constant pressure. After a defined saturation phase (fixed p_{sat}, T_{sat}, and t_{sat}), the outlet valve was

Table 1: Overview about the PLA grades used within this study (data taken from ref. (17))

Grade and supposed usage* (data sheet)	Notation	MFR* (g·(10 min)$^{-1}$ (210°C, 2.16 kg)	D-content* (%)	GPC data M_w (10^3 g·mol^{-1}); M_n (10^3 g·mol^{-1}); PDI (–)	ZSV T = 180°C, γ = 5%, 0.1 rad·s^{-1} (Pa·s)
2003D packaging	P_4.3D	6	4.3 (4)	232; 134; 1.73	8,089
3100HP injection molding	IM1_2D	24	<2 (21)	162: 107; 1.51	1,869
3251D injection molding	IM2_1.4D	80	1.4 (4)	116; 78; 1.49	243
3260HP injection molding	IM3_2D	65	<2 (21)	111; 75,5; 1.47	483
4032D film	Fi_2D	7	1.4–2.0 (4)	232; 149; 1.55	5,716
4044D extrusion	X_4D	6	~4 (21)	230; 117; 1.97	7,794
4060D hot sealing	HS_12D	10	12–12.3 (4)	217; 118; 1.84	4,200
7001D injection stretch blow molding	BM_4.4D	6	4.4 ± 0.5 (4)	242; 142; 1.71	8,472
8052D foaming	Fo_4.7D	14	4.7 (4)	178; 109; 1.64	3,457

*Taken from the data sheet or the literature.

opened to initiate foaming. The experiments were performed with changing saturation conditions, which are summarized in Table 2.

Foam density was determined by the buoyancy method using a Mettler Toledo AG245 balance (Columbus, OH, USA) equipped with a special density kit.

2.3 Dataset and statistical analysis

The dataset includes 258 samples and 13 features: PLA identity, D-content, occurrence of crystallinity (amorph-semicryst), weight average molecular weight (M_w), number average molecular weight (M_n), polydispersity index (PDI), zero shear viscosity (viscosity), melt flow rate (MFR), phase transition temperature (T_{phase}), saturation time (time), saturation temperature (temperature), saturation pressure (pressure), and pressure-drop rate. The variable amorph-semicryst originally took the values "amorphous" or "semi-crystalline" but was encoded as 0 and 1, respectively, for appropriate use in ML modeling. Similarly, the pressure drop rate, which originally took the values "low," "medium," or "high," was encoded as 1, 2, or 3, respectively.

All statistical analyses, including ML regression models, were performed using Python codes (Python 3) written using Jupyter Notebook. The scikit-learn library was used for all regressions. In what follows, density is referred to as the target property and the other variables in the dataset are referred to as features.

The correlations between all pairs of variables (features + target property) from the dataset were first calculated to examine the relationships between these variables before regression models were built. Then, the PCA method was applied to the feature portion of the dataset to determine which samples clustered and which variables were more

Table 2: Overview about the saturation conditions of the autoclave foaming experiments

Sample	Saturation temperature (°C)	Saturation pressure (bar)	Saturation time (min)
P_4.3D	60–130	180	30
IM1_2D	100–165	180	30
IM2_1.4D	100–160	180	30
IM3_2D	100–160	180	30
Fi_2D	80–135	180	7.5, 15, 30, 60
X_4D	40–135	120, 150, 180	7.5, 15, 30, 60
HS_12D	35–130	120, 150, 180	7.5, 15, 30, 60
BM_4.4D	60–140	150, 180	30
Fo_4.7D	80–135	150, 180	30

important in explaining the groups formed. The PCA technique calculates new “principal component” axes that are linear combinations of the features and are intended to contain a large portion of the information (or variance) in the dataset. PCA analysis is used, among other things, to select samples and features for building regression models. Correlations between the major principal components, i.e., PC1, PC2, etc., and the target property were examined to determine if the latter could be predicted from the coordinates of PC alone.

Several ML regression models were created and compared, aiming to predict the target property based on the features. The following collection of linear and nonlinear regression models were initially used: RF, kernel ridge (KRR), GBR, least squares (LR), Lasso, KNN, and support vector (SVR). The hyperparameters of each model were optimized using a random grid search in which vectors containing all hyperparameters with predefined ranges were randomly generated for each ML model and tested using the leave-one-out cross-validation procedure with 60% of the available samples. The optimized ML models were then tested with the remaining samples as described in the next sections. The theoretical background on the ML models used for the final predictions is described in more detail in Section 2.4.

To evaluate the performance of each model, the leave-one-out cross-validation technique was used in which all features of the training set were first preprocessed to have a mean of zero and a unit variance. In this technique, a new model is trained with all samples (= training set) except one (= test set) and the target property of the one-sample test set is predicted using the trained model and compared to the true (experimental) target property. This procedure is then repeated for all samples in the dataset, creating a new model each time. The performance of each model was quantified by the mean absolute error (MAE) and the square of the Pearson correlation coefficient (R^2) calculated between the predicted and true target properties. The parameter MAE is given by:

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad (1)$$

where y_i and $\hat{y}_i$ are the true and predicted densities of sample i, respectively. In addition, the mean absolute percentage error (MAPE) for the final ensemble models was also calculated. The MAPE error is given by:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{|y_i|} \cdot 100\% \qquad (2)$$

The best individual ML models with hyperparameters already optimized were then combined using an ensemble approach (Figure 1). In the first step (Figure 1, top), there is a list of ML models with hyperparameters already optimized and sorted in ascending order from MAE. In the second step, various ensemble models are created, with ensemble predictions obtained using a weighted average of the individual ML models. The positive coefficients (c_i) are optimized for each ensemble model to minimize MAE between the experimental and ensemble-predicted target property. This minimization is performed using the second-order optimization algorithm Broyden–Fletcher–Goldfarb–Shanno (BFGS) in which the c_i's are varied in the direction of the largest decrease in MAE. The ensemble model with the smallest MAE among all possible ensemble models is then selected, from where the individual ML models and their corresponding weights (c_i's) are stored. The final prediction and the corresponding model can then be referred to as the ensemble prediction and the ensemble model, respectively.

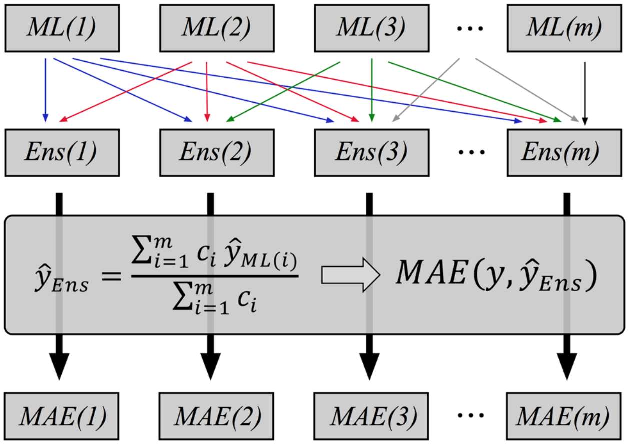

Figure 1: Scheme used to further improve the predictions via an ensemble model. The final ensemble model is the one with the smallest MAE.

The ensemble model was then used to predict the target property for a large number of randomly generated samples (or virtual experiments) to find regions in the feature space associated with optimal (lowest) values of the target property. Note that at this stage, all samples were included in the training of the ensemble model (no sample was omitted). Some of the proposed virtual experiments were selected to be run in the laboratory to further evaluate the performance of the ensemble model.

2.4 Brief description of the ML models

The ML methods listed below are well described in references (20,21) and the specific references for each method are cited below. The main features of each ML model are summarized here.

2.4.1 RF

This model averages the predictions of many uncorrelated decision trees, each of which considers different (randomly generated) subsets of the features and samples (22). Each decision tree consists of a sequence of simple rules, each based on a single feature. For example, if the first feature f_1 is greater than a threshold T_1, the samples are moved to one region (or child node), otherwise to another region or child node. Each rule or condition increases the purity of a region (or node of the tree), which in the case of classification is generally defined based on the Gini index. The different values of all thresholds (T_i's) used for the different rules are those that maximize the purity of the child nodes, i.e., those that minimize the value of the Gini index of the child nodes. All features are tested individually, and their final Gini index improvements are compared to decide which are the best f_i and T_i parameters to use for the current split. For regression trees, the sum of squared errors (SSE) between the predicted and true target property of all samples is used as a parameter to decide which f_i and T_i to use for splitting the trees: A region is split at a certain threshold that minimizes the SSE of both child nodes. If a given node is to be split using feature f_1 and threshold T_1, the predicted target property y for all samples in subregions $R1$ and $R2$ is given by:

$$\hat{y}_{R1} = \text{ave}(y_i|x_i \in R1) \tag{3}$$

$$\hat{y}_{R2} = \text{ave}(y_i|x_i \in R2) \tag{4}$$

where the average ("ave") of the target properties of all samples x_i within the same region is taken to determine the final prediction for that region. Finally, after calculating the SSE for both regions using the corresponding predicted and true target property and varying T_i and f_i, the node is split into child nodes according to the optimized rule and the whole process is repeated until, for example, there are only five samples left per node or a minimum value of SSE has been reached. After all uncorrelated trees are grown, the predicted target property of any sample is calculated by simply averaging the predictions for that sample using all trees. RF is an ensemble model that can perform classification or regression.

2.4.2 GBR

This method uses the gradient descent technique to add new estimators (in this case, regression trees) one at a time to create an optimized ensemble model (23). The regression trees can be grown as described above. After the first tree (considered here as a weak estimator or learner) is grown, the second is added to improve the current model. In this process, the internal parameters of the new tree (e.g., depth of the tree, number of nodes or leaves, etc.) are varied according to a gradient descent procedure to minimize the residual loss of the model. Each new regression tree added to the model in turn has different parameters specifically optimized to reduce the loss of the entire ensemble model. All previously added trees are no longer optimized once they are finally added to the model. One stops adding new trees once the loss reaches an acceptably small value or stops decreasing. The final prediction is obtained by averaging the predictions of all trees in the model. This greedy algorithm makes it possible to greatly increase the accuracy of the model and is a very powerful regression technique, as will be shown later.

2.4.3 KRR

This method uses an L2 regularization term and the so-called kernel trick to make predictions (24,25). Regularization means that larger weight coefficients (w_j) in the linear combinations of features are penalized more than smaller ones. With L2 regularization, the penalty is proportional to the square of w_j. For n samples and m features, the loss function L using L2 regularization is given by:

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{m}w_j^2 \tag{5}$$

where λ is a hyperparameter that relates to how important the L2 regularization is and can be adjusted by cross-validation. Ridge regression has a closed form for calculating w, shown below in matrix form:

$$w = (\lambda I_n + X^T X)^{-1} X^T y \tag{6}$$

where X and y are the features and the target property, respectively, of all samples in the dataset, and I_n is the identity matrix. The superscripts T and -1 represent the transposed and inverted forms of the corresponding matrices, respectively.

The kernel trick expands the original features as linear combinations of potentially infinite terms that are individually unknown, but whose inner product is known, allowing very efficient predictions of the target property for medium-sized datasets. According to the kernel ridge method, the predicted target property y_{new} of any new sample (x_{new}) with an arbitrary number of features is calculated as follows:

$$\hat{y}_{\text{new}} = \kappa(x_{\text{new}})^T(\lambda I_n + K)^{-1}y \tag{7}$$

where K is the full kernel matrix encompassing all pairs of samples (x_i, x_j) in the dataset, and $\kappa(x_{\text{new}})$ is a column vector kernel with all possible pairs of samples (x_i, x_{new}). The matrix elements of both κ and K can be easily computed by assuming one of the many available kernels. For example, for a radial basis function kernel, the matrix element comprising samples a and b is given by

$$K(x_{\text{a}}, x_{\text{b}}) = \exp\left(-\frac{||x_{\text{a}} - x_{\text{b}}||^2}{2l^2}\right) \tag{8}$$

where l is the length scale parameter. The above kernel is a measure of the similarity between the features of samples a and b. In this sense, the full kernel matrix K represents all pairwise similarities between the samples in the dataset.

2.4.4 LR

This method finds a linear combination of the features that minimizes the sum of squares of the errors between the true and the predicted target property. By default, the LR model has no regularization term and is one of the simplest models to build. The weighting coefficients (w_{LR}) of the linear combination are found by minimizing the loss function (mean squared error), resulting in the following expression (in matrix form):

$$w_{\text{LR}} = (X^TX)^{-1}X^Ty \tag{9}$$

where the bias is zero, assuming that the samples were preprocessed to a mean of zero. The calculated weighting coefficients corresponding to the m features are related to the predicted target property of the sample x_{new} by the expression:

$$\hat{y}_{\text{new}} = \sum_{j=1}^{m} w_j x_{\text{new},j} \tag{10}$$

where $x_{\text{new},j}$ is the jth feature of sample x_{new} and w_j is the jth weighting coefficient of w_{LR}.

2.4.5 Lasso regression

This is basically an LR model with an L1 regularization term that penalizes large weighting coefficients via a term that is linear on the weighting coefficients themselves (26). For n samples and m features, the loss function L of Lasso regression can be given as follows:

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{m}|w_j| \tag{11}$$

where the variables and parameters have the same meaning as in the model LR. The L1 regularization causes some weight coefficients to become zero, which means that Lasso regression performs feature selection and can be used with sparse datasets. In addition, the minimization of the Lasso loss function does not have a closed form, so iterative methods are required to find the best w_j's. Prediction of the target properties can be done in the same way as for the LR model (Eq. 10).

2.4.6 KNN

The predictions are based on the similarity between data points, which is often calculated using the simple Euclidean distance between them (27). This method is very efficient and is considered nonparametric since no real training is required, only the computation of the distances between data points. The final expression for the (posterior) probability p_{c} that a new sample x_{new} belongs to class C, which follows from Bayes' theorem, is given by

$$p_{\text{c}}(x_{\text{new}}) = \frac{K_{\text{C}}}{K} \tag{12}$$

where K_{C} and K are the number of samples of class C and the total number of samples within the volume containing the K closest neighbors of x_{new}, respectively. The probability p can be calculated for all available classes. The final class predicted for sample x_{new} results from the majority vote of all K-nearest samples, which also corresponds to the class with the largest p-value. The optimal value of the hyperparameter K is determined by cross-validation as the best value that minimizes the loss function. In the regression, the predicted target property of a new sample is calculated as the average or weighted average of the target properties of x_{new}'s K-nearest neighbors. In the case of the weighted average, the weighting coefficients are usually the inverse of the distance between the K neighboring samples and x_{new}.

2.4.7 SVR

This method optimizes the decision boundaries around the data points using the so-called support vectors, which are defined by the features of certain data points in the dataset (28). The general idea of SVR is to find the flattest possible hyperplane (defined by the weighting coefficients w_j and the bias b) passing through most of the samples in the dataset, where the maximum acceptable deviation from the target property is given by the positive parameter ε: most of the samples are therefore inside a multidimensional ε-tube (also called an ε-insensitive tube). Samples, whose predicted target properties differ by less than ε from the corresponding true target properties, are not included in the loss function and do not penalize the model. Samples that fall outside the ε-tube, also called outliers, are explicitly included in the loss function via the slack variable ξ. The slack variable is the positive difference between the predicted target property of the outlier and the maximum allowable deviation of the predicted target property represented by the walls of the ε-tube. The inclusion of ξ transforms the loss function into a soft margin loss function. The flatness of the hyperplane implies that the weighting coefficients must be small with respect to the features, which is achieved by L2 regularization. The SVR problem for a dataset with t outliers can then be formulated as a minimization of the soft margin loss function

$$\frac{1}{2}\,||w||^2 + C\sum_{i=1}^{t}(\xi_i + \xi_i^\star) \tag{13}$$

over the elements of w while satisfying the three constraints

$$y_i - \hat{y}_i \le \varepsilon + \xi_i \tag{14}$$

$$\hat{y}_i - y_i \le \varepsilon + \xi_i^\star \tag{15}$$

$$\xi_i^\star, \xi_i \ge 0 \tag{16}$$

In the above equations, C is a constant that refers to the tradeoff between the amount to which deviations larger than ε are tolerated and the flatness of the hyperplane. ξ and $\xi^\star$ are the slack variables used when the predicted target properties of the outliers are below and above the ε-tube, respectively. Solving the above equations using Lagrange multipliers leads to an expression for the prediction that is independent of w. SVR can also use the kernel trick described earlier, which leads to the following expression to predict the target property of a sample x_{new} from an SVR model trained using n samples:

$$\hat{y}_{\text{new}} = \sum_{i=1}^{n}(a_i - a_i^\star)K(x_{\text{new}}, x_i) + b \tag{17}$$

where a_i and $a_i^\star$ are the Lagrange multipliers related to the terms with ξ_i and $\xi_i^\star$, respectively, and K is the kernel function (e.g., Eq. 8) evaluated between the two samples inside the parentheses.

3 Results and discussion

3.1 Correlations between variables

Figure 2 shows the visual representation of the correlation matrix with the pairs of the individual variables for all foamed samples of the nine different PLA grades.

Positive correlations (red) indicate that on average, when one variable increases, the other also increases, while negative correlations (blue) indicate that on average, when one variable increases, the other decreases. The correlations are in the range [−1,1] and equal to 1 when they affect a property twice (main diagonal of Figure 2). Note that at this stage no model has been built and Figure 2 only shows the general trends of the whole data.

For example, the variable “amorphous-semicryst” represents the presence of crystals in the samples and is encoded as 0 for amorphous systems and 1 for semicrystalline systems. In Figure 2, the increase in D-content is associated with the decrease in crystallinity (correlation = −0.97), implying that amorphous PLAs tend to have higher D-content, which is consistent with the results in the literature (12). Negative correlations are also found between MFR and the variables M_w (−0.93), M_n (−0.83), and PDI (−0.66). Positive correlations are found between viscosity and the three variables just mentioned: M_w (0.89), M_n (0.82), and PDI (0.58), suggesting that viscosity and MFR should be negatively correlated with each other, as indeed observed in Figure 2 (correlation = −0.76).

According to Figure 2, the two largest correlations involving the target property are found to be MFR (0.49) and M_w (−0.46), indicating that low-density PLA foams can be obtained for smaller MFR and larger M_w values. Figure 2 gives a rough idea of how certain property pairs change together. However, it does not show how a particular property correlates simultaneously with a group of properties or how PLA systems are grouped according to the values of all their processing parameters. This task is better solved by the PCA technique, which is presented in the next section.

3.2 PCA

The relationship between the properties of the whole dataset (all PLA systems) was investigated using the PCA technique, as described in the methodology. The first two principal components, PC1 and PC2, explained 42 and 26% of the variance in the dataset, respectively.

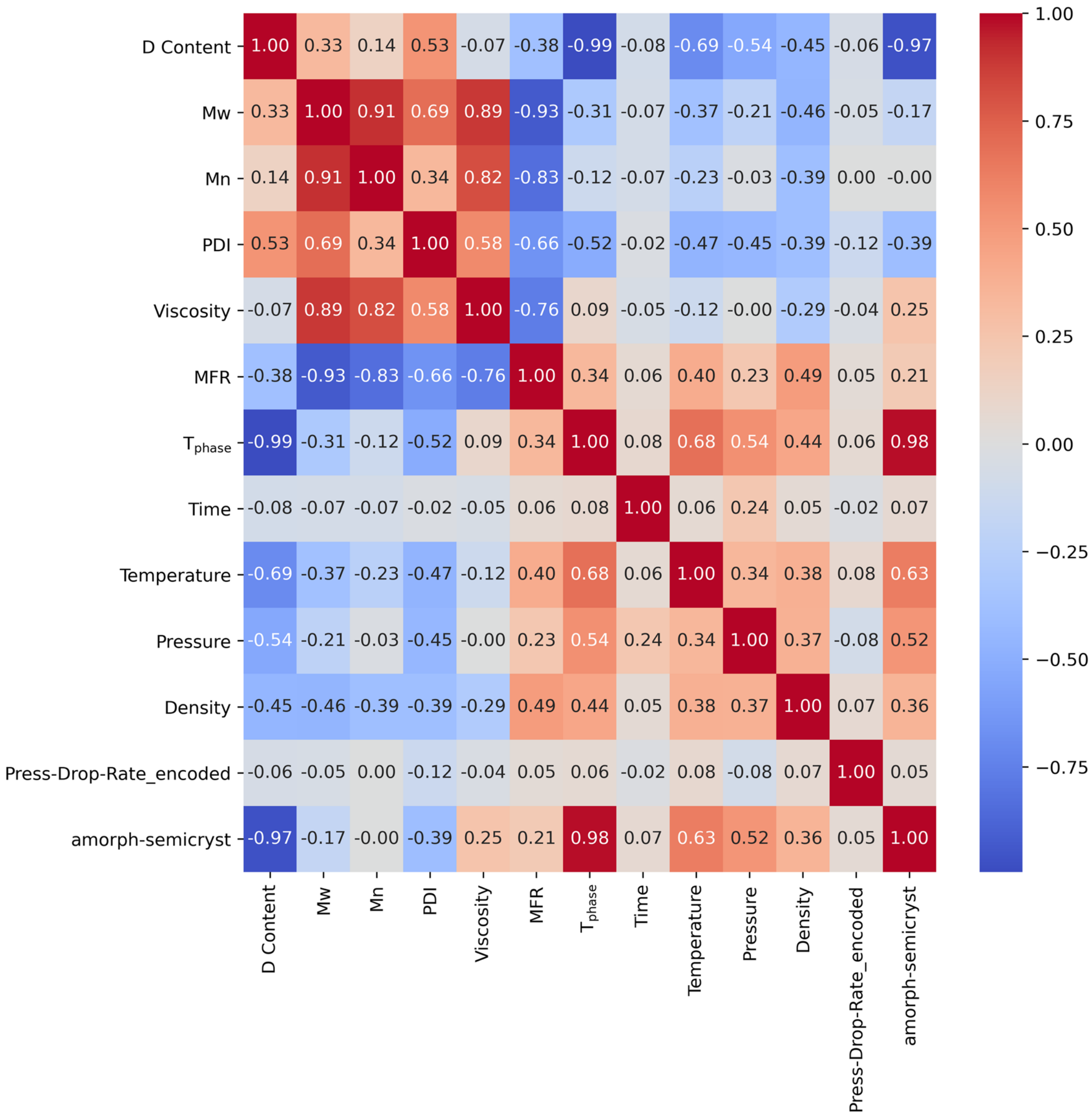

Figure 2: Heatmap table with correlations between every pair of variables (features and target are included) of all samples in the dataset. Red and blue cells represent positive and negative correlations, respectively.

Comparison of PC1 and PC2 (Figure 3) shows that all PLA samples can be classified into five different subgroups, here referred to as A (HS_12D), B (X_4D, BM_4.4D, P_4.3D, and Fi_2D), C (IM3_2D and IM2_1.4D), D (IM1_2D), and E (Fo_4.7D).

The linear combinations used in each principal component can be seen on the respective axes in Figure 3 and provide information about hidden trends in the data. The variables “Time” and “Press-Drop-Rate_encoded” are of little importance for both PC axes because their coefficients in the linear combinations are very close to zero and therefore, they cannot be used to distinguish the PLA groups in Figure 3.

All HS_12D samples (= subgroup A) fall in the range of very positive values of PC2, implying high D-content and low viscosity and M_n. This subgroup also falls in the range of very negative values of PC1, which means a high D-content, M_w, and PDI and a small MFR and T_{phase}.

The samples belonging to subgroup B fall in a region of negative PC2 values, which means that all samples in this group tend to have larger values of variables associated with (or multiplied by) large negative coefficients, such as viscosity and M_n, and smaller values of variables associated with (or multiplied by) large positive coefficients, such as D-content. The PCA analysis correctly grouped the PLA classes based on injection molding (subgroup C).

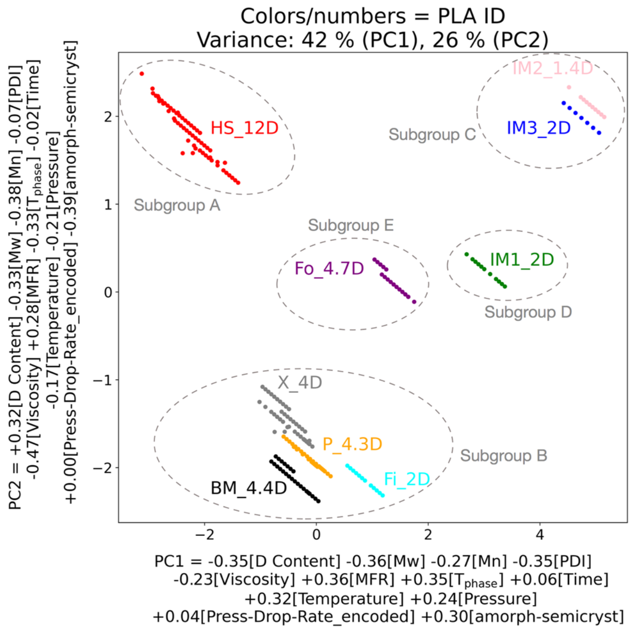

Figure 3: Plot of the principal components PC1 versus PC2 for all the PLA systems investigated, with colored labels representing individual PLAs. The linear combination between variables used in each principal component is also shown.

The target property did not show a pronounced trend in either PC1 or PC2 (Figures S1 and S2 in the supplementary material). For this reason, regression models were created to predict density using the other variables but still using the subgroups generated from the PCA results.

3.3 Regression models

The use of different regression models to predict the density of PLA systems did not show good accuracy when all nine PLA groups were considered simultaneously or for individual subgroups except subgroup A (Figures S3–S5). Here, we discuss two different models including either subgroup A (Case I) or PLAs in subgroups A, C, and D (Case II), as described below.

3.3.1 Case I

For subgroup A, only temperature, pressure, and time were used as features to predict the target property because the other variables were the same.

First, a screening of several linear and nonlinear regression models was performed to determine which models could capture the relationship between the features and the target property. The MAE values (in kg·m^{-3}) for predictions performed with the best individual ML models were 34.1 (GBR), 40.6 (RF), 41.8 (KRR), and 41.9 (SVR) kg·m^{-3}, with GBR outperforming the other models (all optimized ML models are shown in Figure S6). The optimal ensemble model was found by combining these individual ML models and showed significantly better performance than any of the four models on their own.

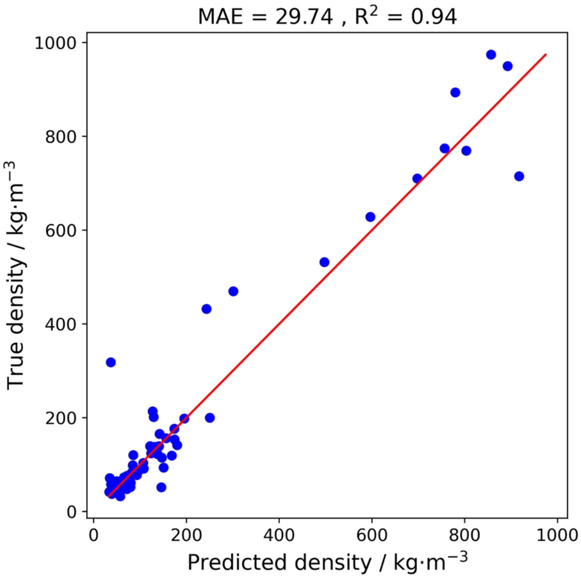

Figure 4: Comparison between the experimental (true) and the ensemble-predicted densities of all HS_12D samples (subgroup A). The ensemble model is the MAE-minimized linear combination of the four individual models (normalized weights in parenthesis): GBR (0.74), RF (0.01), KRR (0.09), and SVR (0.16). The performance shown is based on the leave-one-out cross-validation technique.

The prediction error (MAE) of the optimal ensemble model was 29.7 kg·m^{-3} (Figure 4), which corresponds to a gain of 4.4 kg·m^{-3} compared to the best individual model. Figure 4 shows that as few as three experimental processing parameters (in this case, temperature, pressure, and time) provide satisfactory prediction of the density of HS_12D-based foams. It can also be seen that above about 250 kg·m^{-3} true density, some predictions have significantly larger errors, which is a consequence of the rather small number of samples available for training the ensemble model in this density range. Calculation of the MAPE error for the predictions shown in Figure 4 indicates that the predicted densities deviate on average by 18% from the corresponding experimental densities.

3.3.2 Case II

Samples from PLAs with IDs HS_12D, IM1_2D, IM3_2D, and IM2_1.4D (= subgroups A, C, and D) were combined to create a single ML ensemble model. This was done after the same initial screening of individual ML models and optimization of hyperparameters as previously described for Case I. The choice of these subsets was based on the analysis of the errors of the best ML model created for all PLA samples (Figure S7). Interestingly, all these samples are located in positive regions of PC2 (Figure 3). Here, all variables were used as features in the model. The best individual ML models were GBR (MAE = 48.4 kg·m^{-3}) and KNN (MAE = 57.9 kg·m^{-3}). The minimized ensemble model obtained by combining these two ML models resulted in MAE = 48 kg·m^{-3}, as shown in Figure 5.

The MAE of this ensemble model is still considerably larger than that shown in Figure 4, where a single PLA class was used to train the model. However, in the low-density range, this model is less noisy than the one previously shown in Case I (comparably the MAE of this ensemble model is considerably larger than that shown in Figures 5 for densities below 200 kg·m^{-3}). It appears that using the right combination of PLAs improves the predictions for low densities, which explains why the current MAPE was slightly smaller (15%) than in Case I. Again, the predictions for high densities are much noisier than those for low densities, since for the latter there are fewer data points available to train the ML models.

3.3.3 Screening the processing parameters

The ensemble models shown in Figures 4 and 5 were used to predict the density of randomly generated samples of HS_12D as a function of temperature for various pressures and times to aid in screening the space of processing parameters. The relationship between the predicted density for HS_12D samples as a function of temperature for different pressures and time = 30 min is shown in Figure 6 for the previously described ensemble models in Case I (plots a and b) and Case II (plots c and d).

In all cases, the highest pressure (180 bar) is associated with the largest predicted densities (red curves). In general, the smallest predicted density is associated with a pressure of 140 bar (green curves) in the temperature range of about 75–85°C. Both ML ensemble models showed similar trends, although the second one (Figure 6c and d) shows a much smaller difference between the predicted densities at the minimum of each curve, i.e., the density changes little for different processing parameters.

In Figure 6a and b, the smallest predicted density (18 kg·m^{-3}) was slightly smaller than the smallest experimental density (about 26 kg·m^{-3}). Figure 6 helps to find the best processing conditions for HS_12D foaming by determining the best density for the shortest operating time or for the smallest temperature or pressure values. Similar plots were also created for times other than 30 min (Figures S8 and S9), used in conjunction with Figure 6 to understand how the minimum density predicted for each pressure is affected by time (Figure 7). This provides an additional way to interpret the predictions described above.

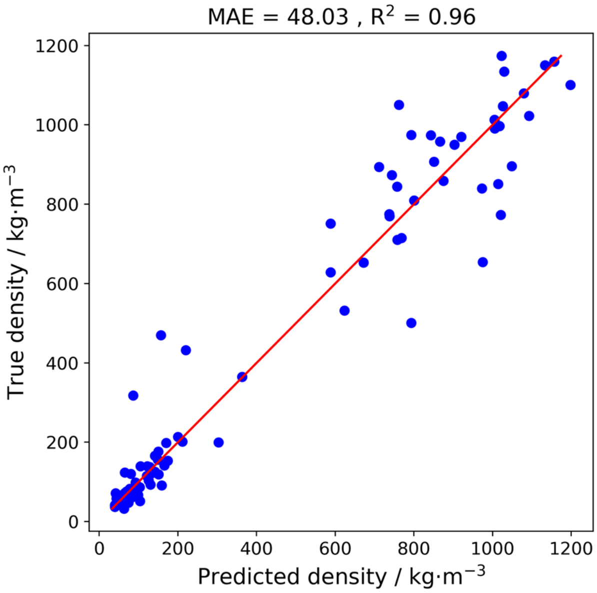

Figure 5: Comparison between the experimental (true) and the ensemble-predicted densities of all samples of the subgroups A, C, and D. The ensemble model is the MAE-minimized linear combination of the two individual models (normalized weights in parenthesis): GBR (0.76) and KNN (0.24). The performance shown is based on the leave-one-out cross-validation technique.

Figure 7 shows the optimal processing times for each pressure, i.e., the processing times that result in the smallest predicted density. Note that the temperatures are not identical for all points shown in Figure 7. Regardless of the ML ensemble model used, the predicted densities become larger for the lowest and highest times. The absolute values of the predicted densities depend on the ML model used, due to the different samples and features used to train the ML models, as described above in Cases I and II. However, most of the trends remain the same.

The available experimental densities as a function of temperature for HS_12D samples at 120 and 180 bar (time = 30 min) are compared with the corresponding predictions performed with the ensemble models of Cases I and II (Figure 8). This confirms that the ensemble model of Case II (green lines/markers) trained with subgroups A, C, and D better describes the low-density regions of the

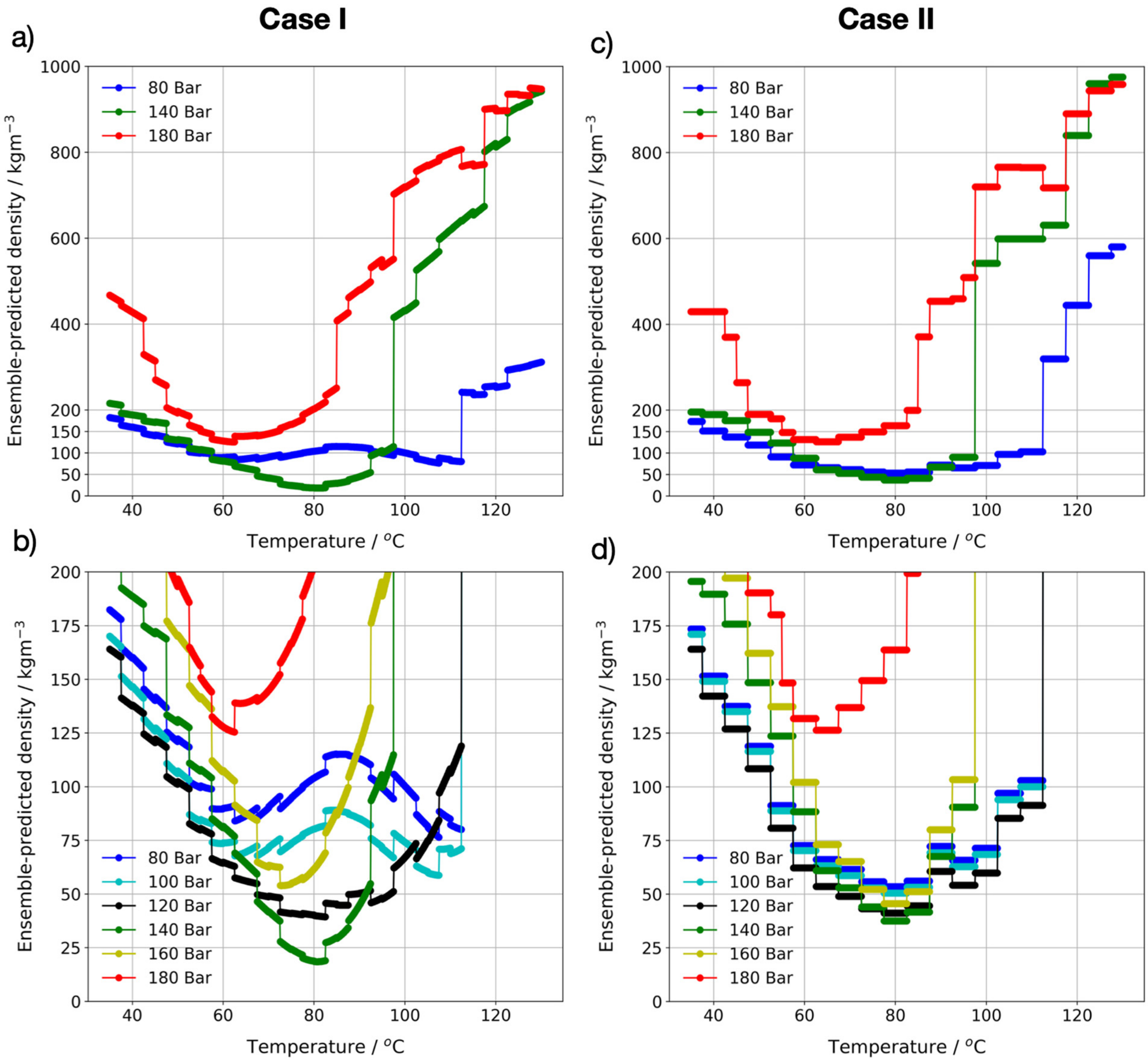

Figure 6: Prediction of density as a function of temperature and pressure (time = 30 min) for randomly generated samples of the HS_12D class. The left column (plots a and b) and the right one were built using the ensemble models previously described in Cases I and II (*vide supra*), respectively. The bottom plots (b and d) have a different *Y*-scale to highlight the low-density region. Some lines of plots (a and c) were omitted for clarity.

HS_12D samples, as already expected from the comparison of Figures 4 and 5. This shows that the inclusion of different PLA classes in the training of the same ML model improves the predictions.

3.3.4 Experimental validation

Based on Figure 6, processing parameters for the production of the new HS_12D foams were proposed. The following parameters, not previously studied experimentally, were chosen: time = 30 min, temperature = 80°C, and pressure = {80, 100, 140, 160} bar. The comparison between these new experimental densities and the corresponding predicted densities is shown in Table 3.

The measured densities for the new PLA foams agreed well with the values predicted by the ML ensemble model discussed in Case II (Table 3). The experimental validation, together with the previous results, makes it clear that one should include different PLA classes in the training of the ML models to make predictions for low densities. One might expect better accuracy from a model based on the material used for verification (Case I). However, the model in Case II shows a better fit to our measured data,

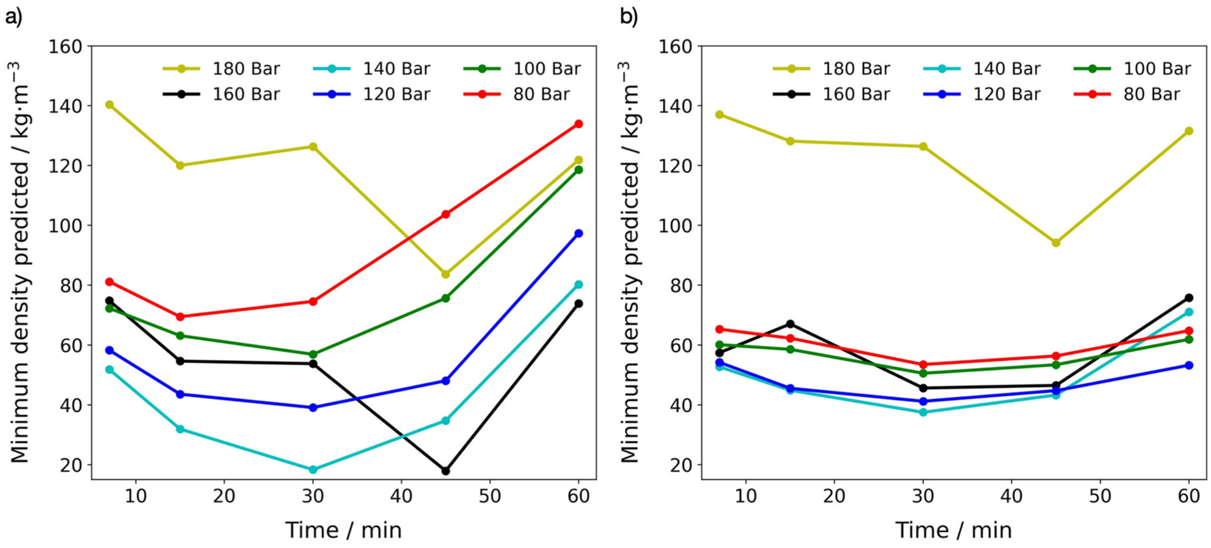

Figure 7: Minimum density as a function of pressure and time predicted for HS_12D samples using the ensemble models previously described in Case I (a) and in Case II (b). The temperature is not the same for all points.

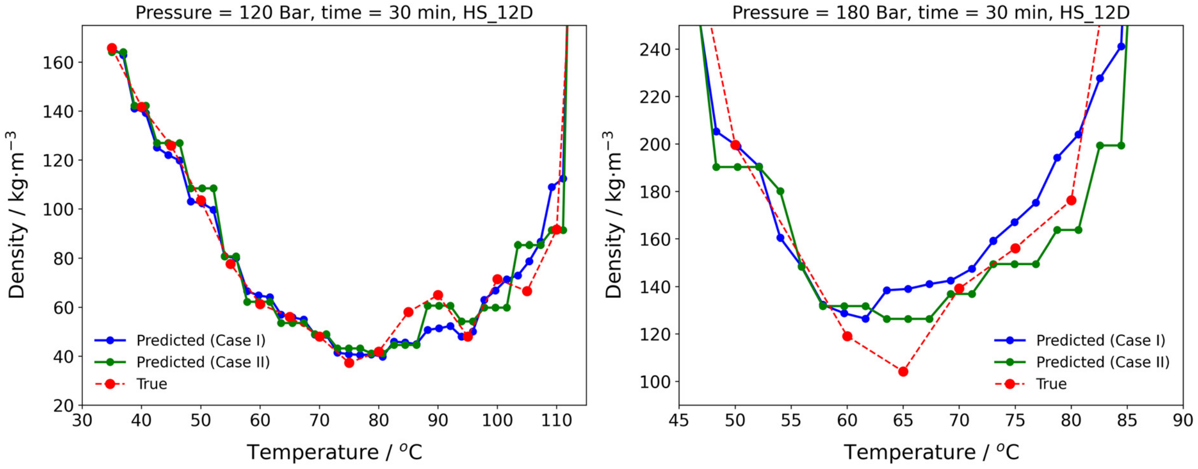

Figure 8: Comparison of experimental and predicted densities of HS_12D samples at two different pressures (120 and 180 bar) and time = 30 min. Predictions were performed using the ensemble models discussed in Cases I and II.

Table 3: Comparison between experimental and predicted densities of new HS_12D foams

Processing parameters (min, °C, bar)	Experimental density (kg·m^{-3})	Predicted density (Case I) (kg·m^{-3})	Predicted density (Case II) (kg·m^{-3})
30, 80, 80	48	104 (117)	54 (13)
30, 80, 100	42	80 (90)	51 (21)
30, 80, 140	36	19 (47)	38 (6)
30, 80, 160	37	61 (65)	46 (24)

The predicted densities were calculated using the ML ensemble models described in Cases I and II. The processing parameters are the time, temperature, and pressure. The percentual errors of the predictions are shown in parenthesis.

which is due to the higher variance of the more diverse dataset consisting of subgroups A, C, and D used to train the model. In addition, the model in Case II takes into account all the properties of the materials and has more data points, which also increases its reliability.

4 Conclusion

In the present work, we have shown a combined experimental–theoretical investigation to develop novel PLA foams with specific densities with a minimal number of new experiments. The linear combination of the best pretrained ML models was used to build the final ML ensemble model, which performed well in predicting the density of HS_12D foams (MAE = 30 kg·m^{-3} and R^2 = 0.94). The results show that combining different PLA classes to train the same ML model is a good strategy to predict low-density values for a particular PLA foam from a grade with a high D-content (HS_12D), compared to training the ML model with only one PLA class. The ML ensemble model was successfully used to identify regions of low density in the theoretical curve of density as a function of temperature at fixed time and fixed pressure. The new processing parameters extracted from the ML-based screening curves were used in the laboratory to produce novel HS_12D foams. The measured densities of the new foams agreed well with the theoretical values of the ML ensemble model described in Case II (exp/theo = 48/54, 42/51, 36/38, and 37/46). These results suggest that the experimental–theoretical approach discussed here can indeed aid in material discovery and improve sustainability. The research described in this article opens up new possibilities for the development of more sustainable methods for the production of PLA foams and could be applied to other materials, processing and analytical methods.

Acknowledgments: We acknowledge all students that were involved in carrying out the experimental trials. Furthermore, we thank the colleagues at the department for their input. The authors thank German Research Foundation (DFG) for funding this project with grant number AL474/34-1. Also, we acknowledge the support of Bavarian Polymer Institute (BPI).

Funding information: The authors thank German Research Foundation (DFG) for funding this project with grant number AL474/34-1. Also, we acknowledge the support of Bavarian Polymer Institute (BPI).

Author contributions: Holger Ruckdäschel: conceptualization, supervision, writing – reviewing, funding acquisition, and resources; Christian Brütting: performing additional autoclave trials and data collection (investigation, data curation, and formal analysis) and writing – original draft; Rodrigo Q. Albuquerque: writing original draft and ML modeling (software, validation, and visualization); Tobias Standau: performing initial autoclave trials and data collection (investigation and data curation), writing – original draft and writing – editing.

Conflict of interest: Authors state no conflict of interest.

Data availability statement: All data generated or analyzed during this study are included in this published article and its supplementary information files.

References

(1) Gartner TE, Jayaraman A. Modeling and simulations of polymers: a roadmap. Macromolecules. 2019;52:755–86. doi: 10.1021/acs.macromol.8b01836.

(2) Kimmig J, Zechel S, Schubert US. Digital transformation in materials science: a paradigm change in material's development. Adv Mater. 2021;33:2004940. doi: 10.1002/adma.202004940.

(3) Dorgan JR, Lehermeier H, Mang M. Thermal and rheological properties of commercial-grade poly (lactic acid)s. J Polym Environ. 2000;8:1–9. doi: 10.1023/A:1010185910301.

(4) Vink ETH, Davies S. Life cycle inventory and impact assessment data for 2014 ingeo® polylactide production. Ind Biotechnol. 2015;11:167–80. doi: 10.1089/ind.2015.0003.

(5) Saeidlou S, Huneault Ma, Li H, Park CB. Poly(lactic acid) crystallization. Prog Polym Sci. 2012;37:1657–77. doi: 10.1016/j.progpolymsci.2012.07.005.

(6) Hartmann MH. High molecular weight polylactic acid polymers. In: Kaplan DL, editor. Biopolymers from renewable resources. Berlin Heidelberg: Springer; 1998. p. 367–411.

(7) Jamshidi K, Hyon S-H, Ikada Y. Thermal characterization of polylactides. Polymer (Guildf). 1988;29:2229–34. doi: 10.1080/00222338508063345.

(8) Dorgan JR, Janzen J, Clayton MP. Melt rheology of variable L-content poly(lactic acid). J Rheol (N Y N Y). 2005;49:607–19. doi: 10.1122/1.1896957.

(9) Ahmed J, Zhang JX, Song Z, Varshney SK. Thermal properties of polylactides: effect of molecular mass and nature of lactide isomer. J Therm Anal Calorim. 2009;95:957–64. doi: 10.1007/s10973-008-9035-x.

(10) Chen J, Yang L, Mai Q, Li M, Wu L, Kong P. Foaming behavior of poly(lactic acid) with different D-isomer content based on supercritical CO2-induced crystallization. J Cell Plast. 2021;57:675–94. doi: 10.1177/0021955X20950242.

(11) Péter T, Litauszki K, Kmetty Á. Improving the heat deflection temperature of poly(lactic acid) foams by annealing. Polym

Degrad Stab. 2021;190:109646. doi: 10.1016/j.polymdegradstab.2021.109646.

(12) Standau T, Zhao C, Murillo Castellón S, Bonten C, Altstädt V. Chemical modification and foam processing of polylactide (PLA). Polymers (Basel). 2019;11:306. doi: 10.3390/polym11020306.

(13) Okolieocha C, Raps D, Subramaniam K, Altstädt V. Microcellular to nanocellular polymer foams: progress (2004-2015) and future directions – a review. Eur Polym J. 2015;73:500–19. doi: 10.1016/j.eurpolymj.2015.11.001.

(14) Standau T, Nofar M, Dörr D, Ruckdäschel H, Altstädt V. A review on multifunctional epoxy-based Joncryl® ADR chain extended thermoplastics. Polym Rev. 2021;63:1–55. doi: 10.1080/15583724.2021.1918710.

(15) Standau T, Murillo Castellón S, Delavoie A, Bonten C, Altstädt V. Effects of chemical modifications on the rheological and the expansion behavior of polylactide (PLA) in foam extrusion. e-Polymers. 2019;19:297–304. doi: 10.1515/epoly-2019-0030.

(16) Brütting C, Dreier J, Bonten C, Altstädt V, Ruckdäschel H. Amorphous polylactide bead foam – effect of talc and chain extension on foaming behavior and compression properties. J Renew Mater. 2021;9:1859–68. doi: 10.32604/jrm.2021.016244.

(17) Standau T, Long H, Murillo Castellón S, Brütting C, Bonten C, Altstädt V. Evaluation of the zero shear viscosity, the D-content and processing conditions as foam relevant parameters for autoclave foaming of standard. Materials (Basel). 2020;13:1–16. doi: 10.3390/ma13061371.

(18) Kuschmitz S, Ring TP, Watschke H, Langer SC, Vietor T. Design and additive manufacturing of porous sound absorbers – a machine-learning approach. Materials (Basel). 2021;14:14. doi: 10.3390/ma14071747.

(19) Mulrennan K, Donovan J, Creedon L, Rogers I, Lyons JG, McAfee M. A soft sensor for prediction of mechanical properties of extruded PLA sheet using an instrumented slit die and machine learning algorithms. Polym Test. 2018;69:462–9. doi: 10.1016/j.polymertesting.2018.06.002.

(20) Bishop CM. Pattern recognition and machine learning (Information science and statistics). New York, USA: Springer; 2006.

(21) Murphy KP. Machine learning: a probabilistic perspective (adaptive computation and machine learning series). Cambridge, USA: MIT Press; 2012.

(22) Breiman L. Random forests. Macine Learn. 2001;45:5–32. doi: 10.1023/A:1010933404324.

(23) Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci. 1997;55:119–39. doi: 10.1006/jcss.1997.1504.

(24) Hoerl AE, Kennard RW. Ridge regression: biased estimation for nonorthogonal problems. Technometrics. 1970;12:55–67. doi: 10.1080/00401706.1970.10488634.

(25) Hofmann T, Schölkopf B, Smola AJ. Kernel methods in machine learning. Ann Stat. 2008;36:1171–220. doi: 10.1214/009053607000000677.

(26) Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc Ser B. 1996;58:267–88. doi: 10.1111/j.2517-6161.1996.tb02080.x.

(27) Altman NS. An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat. 1992;46:175–85. doi: 10.1080/00031305.1992.10475879.

(28) Drucker H, Burges CJC, Kaufman L, Smola A, Vapnik V. Support vector regression machines. Adv Neural Inf Process Syst. 1997;1:155–61.