Polymer Reaction Engineering meets Explainable Machine Learning

Jelena Fiosina[1,*], *Philipp Sievers*[2], *Marco Drache*[2], *Sabine Beuermann*[2, †]

[1] Institute of Informatics, TU Clausthal, Julius-Albert-Str. 4, 38678 Clausthal-Zellerfeld, Germany

[2] Institute of Technical Chemistry, TU Clausthal, Arnold-Sommerfeld-Str. 4, 38678 Clausthal-Zellerfeld, Germany

Abstract

Due to the complicated polymerization technique and statistical composition of the polymer, tailoring its characteristics is a challenging task. Modeling of the polymerizations can contribute to deeper insights into the process. This study applies state-of-the-art machine learning (ML) methods for modeling and reverse engineering of polymerization processes. ML methods (random forest, XGBoost and CatBoost) are trained on data sets generated by an in house developed kinetic Monte Carlo simulator. The applied ML models predict monomer concentration, average molar masses and full molar mass distributions with excellent accuracy (R^2 > 0.96). Reverse engineering results delivering the polymerization recipe for a targeted molar mass distribution are less accurate, but still only minor deviations from the targeted molar mass distribution are seen. The influences of the input variables in ML models obtained by explainability methods correspond to the expert expectations.

Keywords: *polymers, machine learning, kinetic Monte Carlo simulation, multi-target-regression, reverse engineering, explainable AI*

1 Introduction

Polymers are associated with a broad spectrum of properties, which are tailored on demand by selection of the type and conditions of the polymerization process. Up to date, tailoring of

[*] Corresponding author
E-Mail address: jelena.fiosina@tu-clausthal.de (Jelena Fiosina)
[†] Corresponding author
E-Mail address: sabine.beuermann@tu-clausthal.de (Sabine Beuermann)

polymer materials is performed on a sound empirical knowledge complemented by deterministic or stochastic modeling based on chemical and physical understanding of the polymerization process. Currently, the application of machine learning (ML) methods to polymer topics starts to emerge (Martin & Audus, 2023). Radical polymerizations are characterized by a complex reaction mechanism and are difficult to predict, because rather than obtaining distinct substances, these processes yield materials consisting of a large number of macromolecules differing in size and microstructure. In the case of copolymerizations the composition of the polymer molecules and the sequence of incorporated monomer units is affected by the process. Thus, modeling of polymerization processes has to account for all elemental reactions of the polymerization and process specific aspects, e.g., feeding of one or more components over time or the application of temperature profiles. The complexity of detailed polymerization models requires advanced modeling strategies. In addition, tailoring of polymer materials requires the knowledge of the correlation between the polymerization process, the polymer architecture, and the polymer properties. Still, this type of information is scarce.

As laboratory experiments are costly and laborious, the amount of available experimental data may be limited. Simulation methods provide a promising alternative to obtain tailor-made polymeric products. As in laboratory experiments, simulation approaches yield concentration profiles. Instead of real polymer molecules virtual polymer molecules or, so-called in-silico polymers are generated. Moreover, simulations give access to microstructural details of the polymers, which are difficult or even impossible to obtain with real polymers. However, simulations of polymerizations leading to complex products is computationally expensive, and hence, time consuming. Moreover, it does not provide a solution for the problem of reverse engineering that allows for prediction of the input parameters (recipe, reaction conditions, and if applicable - dosing strategies) for polymers with tailored properties. Thus, the prediction of material properties with ML methods using the large amount of data already available appears to be a matter of priority. Figure 1 provides an overview on the output of the different approaches.

Data-driven ML-based modeling and optimization techniques like deep learning and ensemble learning algorithms are starting to open up new opportunities for scientists and engineers working on polymerization processes (Liu et al., 2020; Mohammadi et al., 2019; Mohammadi & Penlidis, 2018). In this study, data-driven ML methods are applied to the reverse engineering of polymerization processes (Figure 1, bottom). Accuracy as well as explainability and transparency are two key desiderata for successful predictive models (Nguyen, Yosinski, & Clune, 2015). Solving the desired roundtrip PRE problems effectively, highly accurate, but complex ‘black-box’ ML models are applied and coupled with corresponding explainability methods. The goal is to bridge the gap between state-of-the-art ML methods and their application in modeling and optimization in polymer reaction engineering (PRE). This strategy constitutes a novel approach referred to as roundtrip PRE, covering polymerization process modeling and reverse engineering of the polymerization process.

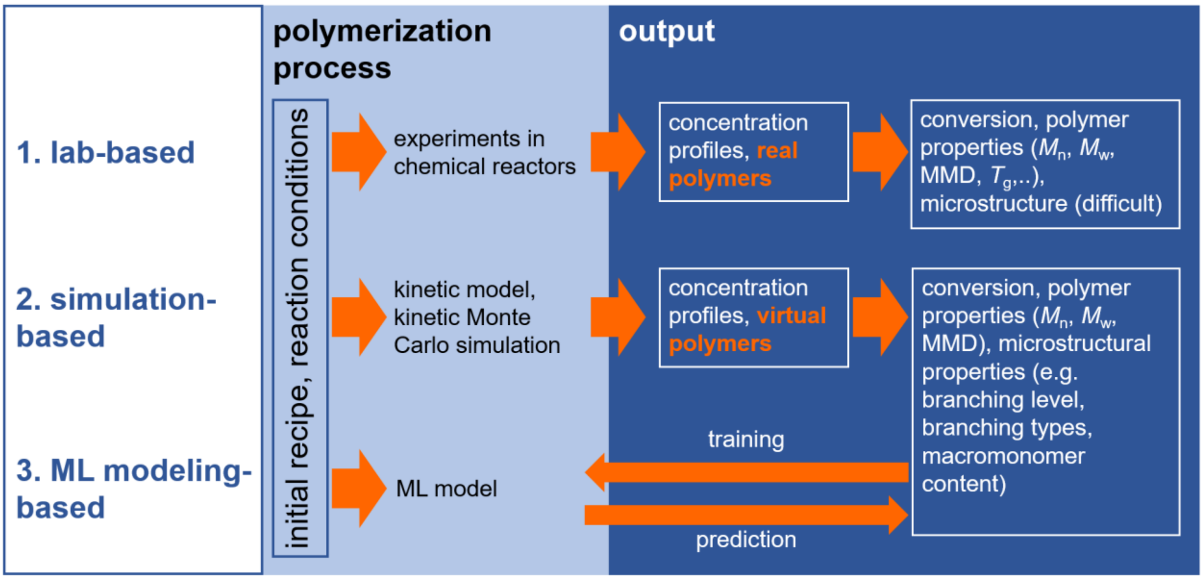
polymerization process
output
1. lab-based
2. simulation-based
3. ML modeling-based
initial recipe, reaction conditions
experiments in chemical reactors
concentration profiles, real polymers
conversion, polymer properties (M_n, M_w, MMD, T_g,..), microstructure (difficult)
kinetic model, kinetic Monte Carlo simulation
concentration profiles, virtual polymers
conversion, polymer properties (M_n, M_w, MMD), microstructural properties (e.g. branching level, branching types, macromonomer content)
training
ML model
prediction

Figure 1: Overview on polymerization process approaches, either lab-based, simulation-based or ML modeling-based, and outputs of the approaches.

2 State of the art

2.1 Polymerization process modeling (PPM)

In the early 2000s, some preliminary supervised learning data-driven approaches towards polymerization process modeling (PPM) based on neural networks with very simple structure and a small number of neurons were reported (Curteanu, 2004; Fernandes & Lona, 2005; Zhang & Pantelelis, 2011). Mohammadi et al. established an ML model (Mohammadi et al., 2019), which is based on deep learning and uses a kinetic Monte Carlo (kMC) simulator in an off-line mode. Recently, Li et al (Li et al., 2018) used deep learning models (e.g., convolutional neural networks) for PPM, which allow for more accurate predictions. Fitting of ML models requires a large training data set, which are hardly available; to address this challenge, Mohammadi et al. propose combining kMC simulation with ML (Mohammadi et al., 2019): results of the kMC simulator are subsequently modelled with data-driven empirical models such as neural networks. The authors report a dramatically reduced computation time compared to the pure kMC simulation approach (Mohammadi et al., 2019). Experimental results are required for testing and cross-validating neural network models to ensure satisfactory predictions (Fernandes & Lona, 2005). Li et al. (Li et al., 2018) reported that after simulation-supported training, the results of such ML models can be used for design of polymerizations in laboratories, instead of real chemical experiments.

Stacked neural networks avoid having to select a single best model; they combine multiple neural networks, which can improve overall representation and robustness (Fernandes & Lona, 2005; Liu et al., 2020). In PPM, taking into account the dynamic aspects of the process yields a number of potential outcomes. Different ensemble techniques were used on top of stacked ML methods (e.g., neural networks) to optimize the final output. Recurrent neural networks can add dynamics enabling real-time management of polymerization processes, provided that the network output is dependent on both its inputs and its prior outputs. In the glass transition temperature and other properties of polymers were evaluated and the performance of ensemble-learning methods (e.g., random forest, XGBoost), neural networks and other regression methods was compared. Various ML methods (including random forest and XGBoost) were applied to predict the conversion and molar mass distribution using multi-target regression (Curteanu, Leon, Mircea-Vicoveanu, & Logofătu, 2021; Da Tan et al., 2022; Dall Agnol, Ornaghi, Monticeli, Dias, & Bianchi, 2021; Ghiba, Drăgoi, & Curteanu, 2021).

2.2 Reverse engineering of polymerization processes (REPP)

The main challenge in training reverse engineering models is that, in contrast to a one-to-one link between polymerization variables and microstructural properties, they may yield many solutions (Mohammadi et al., 2019). A PPM model predicts the polymer properties from the input variables, but inverse modeling is not as straightforward and optimization techniques are required. According to Mohammadi and Penlidis, the reaction condition search space should be intelligently explored in order to identify the best input values for systems with complex reaction mechanisms that produce pre-defined reaction outputs (such as pre-set conversion, yield, and/or other product properties) (Mohammadi & Penlidis, 2018). Genetic evolutionary algorithms are commonly used for optimization in different domains, e.g., for learning fuzzy classifier systems (Afanasyeva, 2002) or for training neural networks by coding unknown adjustable network parameters as chromosomes (Mohammadi et al., 2019). For reverse engineering of polymerization process (REPP), different ML-based optimization techniques were discussed. Mohammadi et al. propose a genetic algorithm-based optimizer that creates a variety of polymerization recipes at random and delivers those recipes to the kMC simulator for error evaluation (Mohammadi et al., 2019). Their optimizer is represented by a non-dominated Sorting Genetic Algorithm. In contrast to Mohammadi et al. (Mohammadi et al., 2019), the use of ML models for reverse engineering of the polymerization process is proposed. The application of REPP instead of a kMC simulator is advantageous, since the process is expected to be faster and may be modified for more complex output structures. Various global optimization techniques, genetic algorithm, particle swarm, improved ant colony, artificial bee colony and differential evolution algorithms, allow finding alternative conditions in the REPP task (Charoenpanich, Anantawaraskul, & Soares, 2020; Dragoi & Curteanu, 2016; Fernandes & Lona, 2005). As a simple solution, we use ML for initial solution of reverse engineering problems. Furthermore, various optimization methods and heuristics can be applied to find an optimal solution, which satisfies the given criteria, conditions and restrictions.

2.3 ML methods

Currently, ML as a data driven modeling approach does not only provide more accurate solutions for existing problems but also finds novel applications in different domains, e.g., transportation (Fiosina, Fiosins, & Müller, 2013), bioinformatics, medicine, chemical engineering. Multivariate linear regression is one of the simplest ML methods. It is parametric, has naturally explainable coefficients and in case of linear dependencies can provide accurate solutions to a big number of problems (Draper & Smith, 1998). Decision-tree is another simple interpretable ML method, which can be used for forecasting. It uses the classification and regression trees algorithm (CART) proposed by Breiman et al. (Breiman, Friedman, Olshen, & Stone, 1984) to find an optimal decision tree construction.

The bagging technique based on building several decision-tree models on bootstrapped data sets can drastically improve the performance of CART. In 2001 Breiman proposed a random forest algorithm (Breiman, 2001), which builds several decision trees on bootstrapped data sets. Each tree independently predicts the results, and the final solution is obtained by voting or as an average of the individual tree solutions.

The next evolution step in decision-tree based ensemble methods is the application of a boosting technique, in which the models are built sequentially such that each model tries to correct the mistakes made by the previous one. Extreme Gradient Boosting (XGBoost) (Chen & Guestrin, 2016) is a gradient boosting-based technique that Chen proposed in 2016 and is an implementation of gradient boosted decision trees built for speed and performance. Another implementation of the gradient boosting technique is the CatBoost (Prokhorenkova, Gusev, Vorobev, Dorogush, & Gulin, 2018) algorithm, which was proposed by Yandex. This ML method often performs better, especially with categorical input features. Gradient boosting algorithms often are more accurate models, but tend to suffer from overfitting (Rokach, 2019). An effective investigation of various decision-tree based ensemble algorithms applied to the prediction of kickstarter campaigns was considered in (Jhaveri, Khedkar, Kantharia, & Jaswal, 2019). Ensemble-based methods (e.g., random forest, XGBoost, CatBoost) perform well; however, the major drawback of these methods is that the averaged model is no longer easily interpretable. In the next sub-section, we will address this problem in more detail.

2.4 Explainable ML methods for PRE

Complicated ML techniques, such ensemble-based models or neural networks, may learn what happens throughout a process without modeling the underlying physical and chemical laws. Thus, they facilitate modeling of complex non-linear processes with a limited understanding of the phenomenon. The ML technique, however, does not clarify the reaction mechanism itself. The two approaches - simulation and ML - are nicely complementary. The findings from a kMC simulation, which is based on physical and chemical rules, can occasionally be less precise than those from ML models, which describe empirical characteristics (Fernandes & Lona, 2005). As such ML models are ‘black box’ models (Holzinger, 2018), this may render the process non-transparent and not trustworthy for the scientists and engineers, who are users of an ML-supported PPM/REPP system. They need to understand the underlying processes and have need of and rights for explanations; the same is true for the developers of ML-supported systems, who wish to prove model correctness.

There are model-agnostic and model-specific methods to provide insights into its decision-making process of ‘black-box’ models. Model-agnostic methods, like LIME (Ribeiro, Singh, & Guestrin, 2016), Shapley Values (Castro, Gómez, & Tejada, 2009; Lundberg & Lee, 2017), permutation importance (Molnar, 2022) are applicable for different types of models, however, they are very computation-intensive and often are not applicable for big data sets used in deep learning (Molnar, 2022). Model-specific methods tend to be more suitable for specific ML models (e.g., decision-tree based ensemble models, deep learning), which focus on only one type of the models. Examples for such neural network specific explainability methods are Integrated Gradients (Sundararajan, Taly, & Yan, 2017), DeepLIFT (Ancona, Ceolini, Öztireli, & Gross, 2018; Shrikumar, Greenside, & Kundaje, 2017).

In this study, the built-in model-specific explainability method ”Mean decrease in Impurity” (Breiman et al., 1984) and the model-agnostic method ”PredictionValueChange” ("CatBoost library," 2023) for decision-tree based ensembles were used and helped to understand the models.

3 Problem statement

For clarity of presentation, throughout the text we will use the term “simulation” for the procedure of obtaining polymer properties by the kMC simulator and “modeling” for obtaining polymer properties by ML methods.

3.1 General Concept

Figure 2 illustrates the general approach of state-of-the-art PRE employed in this investigation, considering PPM and REPP. In PPM, several polymerization recipes and conditions are pre-defined, and the simulations are separately performed for each case using the kMC simulator. Then, the in-silico polymer is analyzed to determine its macromolecular characteristics. The average or distributional properties of well-defined indices precisely quantify the microstructure of the simulated chains. In this study, the focus is on the correlation of the batch polymerization process applying different recipes and temperatures with molar masses of the resulting polymer.

Since simulations are computationally expensive and time consuming, rather than using kMC simulations a suite of ML models for accurate prediction of polymerization properties is developed. Moreover, the proposed ML models are quickly executed and represent an alternative to the kMC simulator in time-consuming problem settings, like reverse engineering. A novel simulation-supported suite of ML-based models is introduced, which allows for polymerization modeling and reverse engineering as illustrated in Figure 2. The approach enables (i) target-oriented production of tailored materials on demand without long downtimes and large quantities of material being off-spec, (ii) design of new sustainable production processes, (iii) development of polymers with better or even new properties, (iv) a better understanding of the detailed reaction mechanisms and the associated kinetic parameters. The main goal is broken down into the following scientific objectives: (1) to create a coherent suite of scalable ML-based models for polymerization modeling facilitating efficient simulation-supported learning of ML models; (2) to create ML-based approaches for reverse engineering of polymerization processes with modeling and optimization capabilities, based on experimental and simulation data; (3) to explain the proposed ML-models.

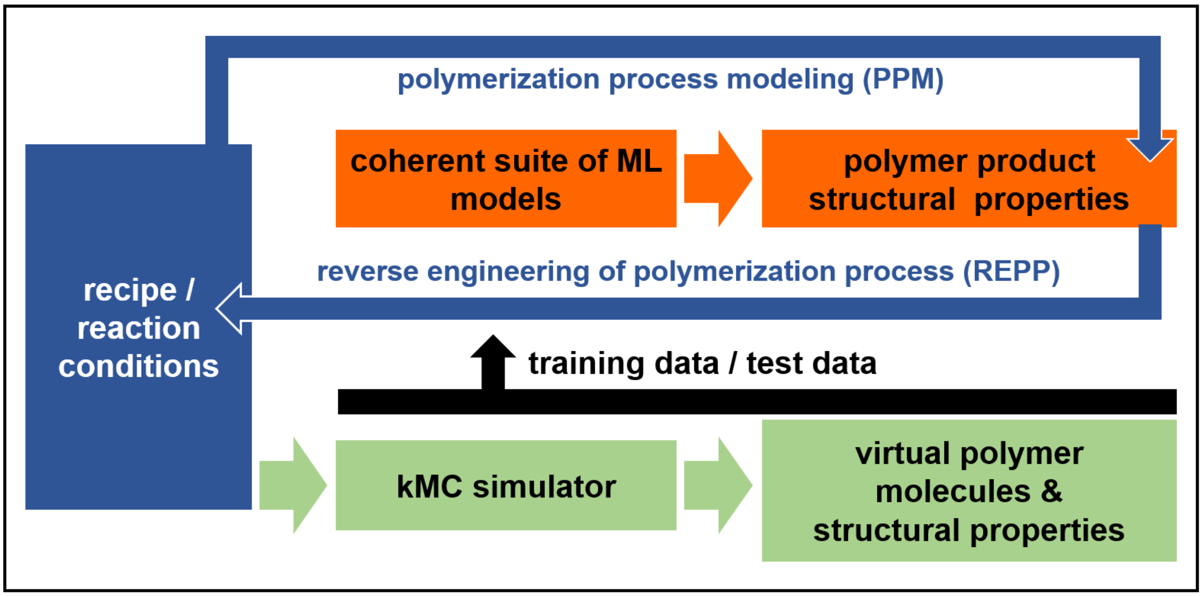
polymerization process modeling (PPM)
coherent suite of ML models
polymer product structural properties
recipe / reaction conditions
reverse engineering of polymerization process (REPP)
training data / test data
kMC simulator
virtual polymer molecules & structural properties

Figure 2: Process of simulation-supported ML-based polymer product development.

3.2 Roundtrip polymerization reaction engineering (PRE)

Above the roundtrip PRE is introduced (Figure 2), consisting of PPM and REPP. In PPM (Mohammadi et al., 2019), the kMC simulator receives a vector $X = (x_1, x_2, \dots, x_m)$ of polymerization variables as input and yields microstructural arrays $Y = (y_1, y_2, \dots, y_d)$ as output. In this modelling approach, the direction is $X \rightarrow Y$. REPP (Fernandes & Lona, 2005; Mohammadi & Penlidis, 2018) is a more complex task, i.e., a search process whereby the results of a system with desired product quality are used to obtain initial operating conditions of polymerization processes. Hence, selecting reaction conditions for desired microstructural properties in optimization of $Y \rightarrow X$, can lead to an ill-posed problem (Mohammadi et al., 2019).

The monomer butyl acrylate (BA) is selected because of its technical relevance and a complex but well-established reaction mechanism, which has been already implemented in kinetic kMC models (Drache, Hosemann, Laba, & Beuermann, 2015; Edeleva, Marien, van Steenberge, & D'hooge, 2021). Thus, a large number of kMC-derived data covering a wide variety of reaction conditions and different degrees of complexity is available. Different ML methods are compared to identify the most accurate for the given setting. The outputs of ML-based models are verified using data generated by the kMC simulator based on metrics such as the coefficient of determination (R^2), mean squared error (MSE), similarity metrics to ensure sufficient performance of the constructed ML models. The kMC simulator allows for interrupting the reaction process at any time to write the current state and to continue reactions afterwards. ML models with multi-target regression (MTR) are considered, because the elements of the output vector representing MMDs for defined molar mass intervals are dependent from each other (Da Tan et al., 2022) and are better predicted taking into account these dependencies rather than using individual single-target models. For the same reason MTR is used for prediction of monomer concentration and average molar masses for different time moments, which are also interdependent.

3.3 Prediction scenarios: Polymer properties and polymerization process characteristics

ML does not allow for learning what happens in the process and does not explicitly model the physical and chemical laws that govern the system. Contrary to in-silico polymers from kMC simulations, the ML models can only predict polymer properties like average molar masses, full MMDs, as well as the monomer concentration. The following prediction models are built for the monomer butyl acrylate. All initial (test and training) data for the ML models are available from the corresponding kMC simulations. Azobisisobutyronitrile (AIBN) is used as initiator and 2-octanone as solvent. However, the solvent concentration is excluded from the ML models, because this parameter is defined by the other components' concentrations.

BA concentration: The variation of BA concentration with time, $c_{\mathrm{BA}}(t)$, is predicted for different cases, in which the initial monomer and initiator concentrations $c_{\mathrm{BA},0}$ and $c_{\mathrm{AIBN},0}$, respectively, as well as the temperature, T, were changed:

$$c_{\mathrm{BA}}(t)=\mathrm{f}\big(T, c_{\mathrm{BA},0}, c_{\mathrm{AIBN},0}\big),\ t=t_1,t_2,\ldots,t_n \qquad (1)$$

M_n and M_w prediction: Polymer molecules consist of repeat units (monomers) linked by chemical bonds to long chains. The molar mass of the polymer chain, which is correlated to the molar mass of the monomers, is frequently used to express chain length. However, the length of polymer chains is disperse. Therefore, the molar mass is not a single value but a distribution. The number average molar mass, M_n, and the weight average molar mass, M_w, are defined as:

$$M_\mathrm{n}=\frac{\sum_i N_i M_i}{\sum_i N_i},\ M_\mathrm{w}=\frac{\sum_i N_i M_i^2}{\sum_i N_i M_i}, \tag{2}$$

where N_i is the number of polymer molecules with the molar mass M_i.

Within the ML models M_n and M_w are calculated as functions:

$$M_\mathrm{n}(t)=\mathrm{f}\left(T, c_\mathrm{BA,0}, c_\mathrm{AIBN,0}\right), \qquad M_\mathrm{w}(\mathrm{t})=\mathrm{f}\left(T, c_\mathrm{BA,0}, \mathrm{c}_\mathrm{AIBN,0}\right), \quad t=t_1, t_2,\ldots,t_n. \tag{3}$$

Molar mass distribution (MMD) prediction: The MMD represents the number of molecules of each polymer species N_i and the corresponding molar mass M_i. The kMC simulator generates the MMD from the chain length distribution. Further, the MMD for each time moment depending on the polymerization recipe and temperature is predicted:

$$MMD(t)=\mathrm{f}\left(T, c_\mathrm{BA,0}, c_\mathrm{AIBN,0}\right) \quad t=t_1, t_2,\ldots,t_n. \tag{4}$$

Note, that $MMD(t)$ returns a vector with predicted $w(\log{(M)})$ values for each interval of the simulated MMD. In this study, we have chosen the number of intervals equal to 100.

Reaction time prediction:

In addition to the above-described quantities, the reaction time, t, is predicted. Generally, it is advantageous to reach full monomer conversion in polymerization processes to avoid the removal of residual monomer. In addition, the reaction time needs to be limited. As a compromise, in this study the reaction time required to reach 90 % of monomer conversion X is targeted:

$$t = \mathrm{f}\left(T, c_\mathrm{BA,0}, c_\mathrm{AIBN,0}\right)_{X=0.9}. \tag{5}$$

MMD Reverse engineering: A model is built for the direct prediction of a polymerization recipe and temperature that leads to a targeted MMD:

$$RE\left(MMD(t)\right)=\left\{T, c_\mathrm{BA,0}, c_\mathrm{AIBN,0}\right\}. \tag{6}$$

This task provides the first insight towards a more complex optimization procedure to obtain an optimal recipe for a targeted MMD. Firstly, the model takes the MMD for a single given reaction time into account, while the optimal conversion of monomer and the minimal reaction time will be addressed in future studies.

4 Data acquisition by simulation

4.1 kMC Simulation

Kinetic Monte Carlo simulations have been proven to be a versatile tool for the stochastic modeling of chemical processes. The algorithm was first published by Gillespie (Gillespie, 1976). Since then improvement in computing power and coding allowed for the application of the algorithm to more and more complex systems (Trigilio, Marien, van Steenberge, & D'hooge, 2020). Today polymer reaction engineering constitutes one very important field of kMC simulations (Brandl, Drache, & Beuermann, 2018; D'hooge, 2018; Drache & Drache, 2012; Hernández-Ortiz et al., 2017; Iedema & Hoefsloot, 2006; Peikert, Pflug, & Busch, 2019; Saldívar-Guerra, 2020; Trigilio, Marien, Edeleva, van Steenberge, & D'hooge, 2022; van Steenberge et al., 2017). Since kMC simulations consider single molecules, it is possible to gain insights into the microstructure of each polymer molecule.

The kMC simulations were conducted using the in-house created open source kMC simulator mcPolymer (Drache & Drache, 2012), which is based on a full kinetic scheme with all elemental reactions occurring. Previously, the simulator was successfully applied to model, e.g., reversible deactivation radical polymerizations (Drache, 2009; Drache & Drache, 2012), acrylate polymerizations with backbiting and transfer to polymer reactions (Drache et al., 2015), or semi-batch vinyl acetate polymerization (Feuerpfeil et al., 2021).

4.2 Training data set acquisition

The training and test data are generated by kMC simulations based on a detailed kinetic model for BA polymerizations including all relevant elemental reactions (Drache et al., 2015). Amongst others, monomer concentrations and MMDs are generated (Figure 3) for technically relevant conditions as discussed in section 6.1. The kMC simulator outputs polymer molecules with complex microstructure, which cannot be directly used to train ML models. For this purpose, the raw data is filtered, abstracted (e.g., MMD), pre-processed, logically connected and stored in the well-structured no-SQL database MongoDB ("MongoDB: The Developer Data Platform," 2023). This interface facilitates data storage from the kMC simulator and provides comfortable and quick access to the training and test data for various ML models, and polymerization products.

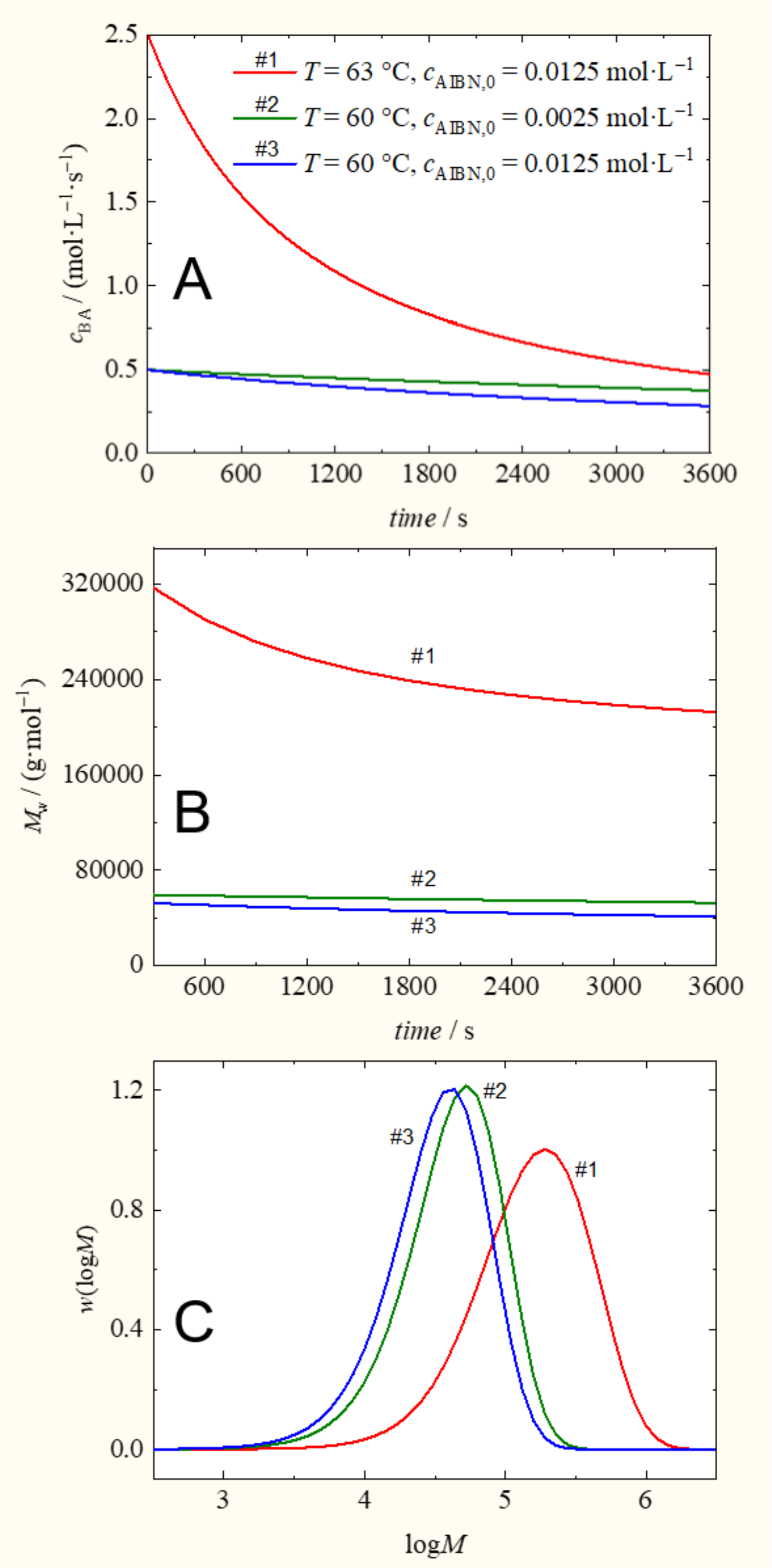

Figure 3: Example results from the kMC simulator for three different starting conditions depicting the variation of c_{BA} and M_{w} with time as well as the MMD at a reaction time of 3600 s.

Moreover, the generation of training, test, and validation data is scalable and fully automatized. In order to minimize the kMC simulation effort, it is investigated how many simulated data is required to obtain satisfactory predictions with the ML models.

5 Models and Methods

5.1 Multi-target regression model

Multi-target regression (MTR) (Spyromitros-Xioufis, Tsoumakas, Groves, & Vlahavas, 2016), allows predicting multiple continuous variables on the base of a common set of input variables. MTR was used in various fields including ecology (Kocev, Džeroski, White, Newell, & Griffioen, 2009), energy (Meyer, 2021), and economics (Ghosn & Bengio, 1996).
Firstly, the MTR problem is formally described and the corresponding notation is provided.
Let the training set S contains N instances with values for each independent variable $X_1, X_2, \ldots, X_m$, and each dependent variable $Y_1, Y_2, \ldots, Y_d$, i.e., $S = \left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}\right), \ldots, \left(\mathbf{x}^{(N)}, \mathbf{y}^{(N)}\right)$, where $\boldsymbol{x}^{(k)} = \left(x_1^{(k)}, \ldots, x_j^{(k)}, \ldots, x_m^{(k)}\right), j \in 2, \ldots, m-1$, and $\boldsymbol{y}^{(k)} = \left(y_1^{(k)}, \ldots, y_i^{(k)}, \ldots, y_d^{(k)}\right)$, $i \in 2, \ldots, d-1$, $k \in 1, \ldots, N$.
MTR fits a model h from S finding (Borchani, Varando, Bielza, & Larrañaga, 2015):

$$\begin{gathered} h\colon \mathbf{\Psi}_{X_1} \times \ldots \times \mathbf{\Psi}_{X_m} \longrightarrow \mathbf{\Psi}_{Y_1} \times \ldots \times \mathbf{\Psi}_{Y_d} \\ \boldsymbol{x} = (x_1, \ldots, x_m) \mapsto \boldsymbol{y} = \left(y_1, \ldots, y_d\right), \end{gathered} \tag{7}$$

where $\mathbf{\Psi}_{X_j}$ and $\mathbf{\Psi}_{Y_i}$ are the corresponding sample spaces.

Alternatively, in the single-target regression a model h is represented as d single-target models h_i. Each model h_i is fitted on a reduced training set $S_i = \left(\boldsymbol{x}^{(1)}, y_i^{(1)}\right), \ldots, \left(\boldsymbol{x}^{(N)}, y_i^{(N)}\right)$, $i \in 1, \ldots, d$, to predict the value of each variable Y_i. In single-target regression, target variables are modeled independently without taking into account potential dependencies between them. The advantages of using multi-output models instead of a combination of single-output models are listed by (Karkera, 2017).
Utilizing MTR for the prediction it is taken into account, which ML methods support multiple outputs. Alternatively, an ensemble of single-target regression models are considered, in which each model predicts the dependent variable for an output separately. Multi-output models as

random forest and CatBoost are compared with single output models like XGBoost that is only capable of producing a series of individual models (Rokach, 2019). One of the research questions here is to investigate, which model provides the best predictions.

5.2 Explainability methods

An advantage of utilizing such ensemble-based decision tree methods as random forest or gradient boosting (XGBoost and CatBoost) is that they automatically provide estimates of feature importance for each input variable. Generally, feature importance indicates how valuable each variable was in the model. The relative importance of a variable depends on how often it is used to make decisions. Variables are ranked by this importance and compared to each other. The importance of each variable of a single decision tree is determined by the amount of improved performance measure resulted by each split point, weighted by the number of observations of the node (Hastie, Tibshirani, & Friedman, 2008). The performance measure may be the Gini index or another more specific error function. Variance is the measure of impurity for regression. The more important the variable is, the more it decreases the impurity. Variables at the top of the tree are in general more important than variables at leaves, as bigger information gains correspond to the top splits. In ensembles of decision trees, the final importance of the variables can be averaged for all trees within the model (Elith, Leathwick, & Hastie, 2008). Another approach to calculate feature importances in CatBoost method is based on the changes of prediction values ("CatBoost library," 2023). It shows how much the prediction changes on average, when the variable value changes.

In this paper, the ensemble models were explained with specific built-in functions ”mean decrease in impurity” (MDI) for the random forest and XGBoost methods (Breiman, 2001) and ”PredictionValueChange” for CatBoost ("CatBoost library," 2023).

The applied ML models use different mechanisms to interpret the influence of input variables. Thus, it is difficult to compare unscaled linear regression coefficients with already normalized decision-tree based ensemble models. To be able to compare the explainability results of linear regression (regression coefficients) and ensemble methods (feature importance), the data for linear regression was standardized, and then, the normalized contribution of its coefficients (sum of the contribution is equal 1) was calculated.

6 Results and discussion

6.1 Generation of training data and experimental design of ML models

The kMC simulations were performed with BA as monomer, AIBN as initiator, and 2-octanone as solvent. The following polymerization parameters were used: T ranging from 60 to 80 °C, $c_{\mathrm{AIBN,0}}$ in the range of 2.5 to 20.0 mmol·L^{-1} and $c_{\mathrm{BA,0}}$ in the range of 0.5 to 3.0 mol·L^{-1} with uniformly distributed grid size, thus resulting in 432 simulations of the process. The polymerization process was simulated for a reaction time of 3600 seconds and the properties of interest were recorded every 300 seconds, thus, obtaining 12 data points for different time moments in total for each investigated property. Obviously, the reaction time was not pre-set in the reaction time prediction model, in which we aimed to reach 90 % monomer conversion, which was reached in some rare situations after about 17 hours. For training the ML prediction models the obtained data set was divided into training and test set in proportion 80:20. The R^2 metric was calculated on the test set data to estimate the performance of the model. A five-fold cross validation was carried out to get the average performance for each prediction models. As stated in section 4, to identify the amount of data required to obtain satisfactory results, the number of data records used for training the ML models was reduced according to the reduction criteria in Table 1. In order to keep a grid while reducing the data set size, the reduction was performed in a methodical way. Firstly, intermediate values for the less important variable $c_{\mathrm{AIBN,0}}$ were dropped, reducing the data set size to 61 %. Secondly, intermediate temperature values were deleted reducing the size of the data set to 34 %. In the third step, intermediate BA concentrations were dropped reducing the data set size to 17 % and further to 10 %.

The tuned hyper-parameters of ML models are presented in Table S1 of the Supporting Information. For the prediction of each property, the following aspects were addressed:

(1) Which of the considered ML models provide the most accurate predictions?

(2) How does the model performance reduce with decreasing the available training data? Which amount of training data is necessary to obtain satisfactory and reliable predictions?

(3) How do polymerization parameters influence the prediction model results? Are the results of explainability methods reasonable from a chemical point of view?

Table 1: Reduction rules for the training data set

average training set size/ %	**reduction rule**
345 / 100 %	full data set, no reduction: $c_{\mathrm{AIBN,0}}$: 2.5 to 20.0 mmol·L^{-1}; T: 60 to 80 °C; $c_{\mathrm{BA,0}}$: 0.5 to 3.0 mol·L^{-1}
218 / 61 %	$c_{\mathrm{AIBN,0}}$ without [7.5, 12.5, 17.5] mmol·L^{-1}
117 / 34 %	$c_{\mathrm{AIBN,0}}$ without [7.5, 12.5, 17.5] mmol·L^{-1}, T without [63, 68, 73, 78] °C
58 / 17 %	$c_{\mathrm{AIBN,0}}$ without [7.5, 12.5, 17.5] mmol·L^{-1}, T without [63, 65, 68, 73, 75, 78] °C, $c_{\mathrm{BA,0}}$ without [2.0] mol·L^{-1}
34 / 10 %	$c_{\mathrm{AIBN,0}}$ without [7.5, 12.5, 17.5] mmol·L^{-1}, T without [63, 65, 68, 73, 75, 78] °C, $c_{\mathrm{BA,0}}$ without [1.0, 2.0, 2.5] mol·L^{-1}

6.2 Reaction time prediction

Equation (5) is used to predict the reaction time required to reach a monomer conversion of 90 %, which is important for successive reverse engineering. Only simulations reaching the 90 % conversion level in less than 10 h were considered. Batch processes with more than 10 h reaction time are considered to be technical unimportant. Also, at high initiator conversions the kMC simulation can become inaccurate due to a low number of radicals. The histogram of the cleaned data set is presented in Figure 4B with time intervals given at the x-axes. As the data are not normally distributed, which is required by linear regression model, we used the logarithmic transformation of the dependent variable for this model only. The results of prediction with four ML methods are presented in Figure 4. Figure 4A shows the simulated and predicted results from the models trained on the full training set for data from the test set ordered by reaction time The deviations of predictions from simulations are relative small, especially for CatBoost and XGBoost methods, but the predictions of linear regression are not as precise. Figure 4D shows that the relative prediction errors for the CatBoost model are the smallest with about 3 % on average. Then, follows XGBoost with about 5 % on average, and random forest, which is slightly less precise. Linear regression returns about 12 % of errors on average.

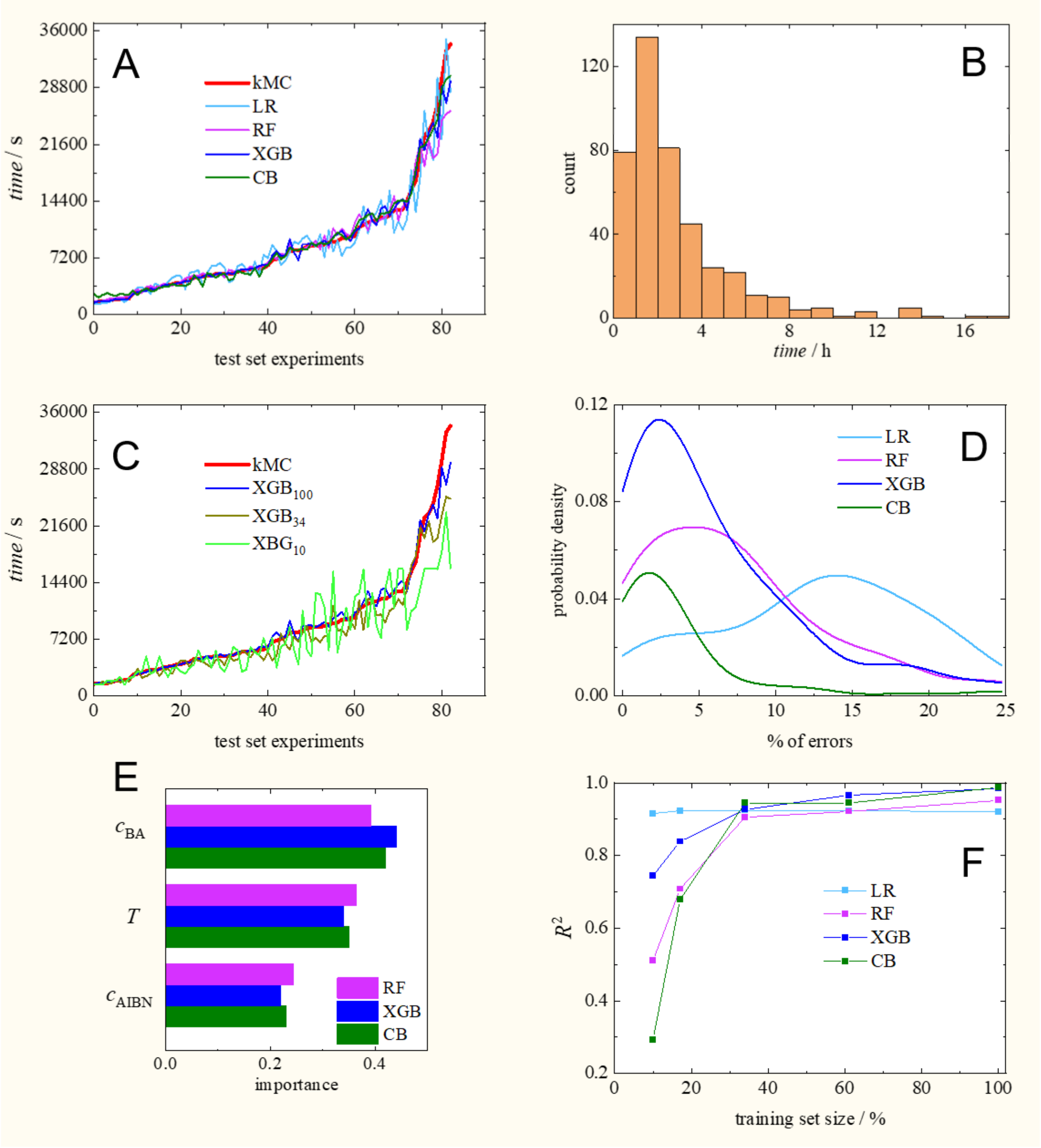

Figure 4: Reaction time prediction models: example test set predictions based on the full (A) and reduced (C) training data sets for the indicated methods, histogram of reaction times (B), distribution of relative prediction errors (D), feature importance (E), and model performance (F).

Figure 4F and Table S2 shows that the best performance for the whole training set was obtained with the CatBoost regression model, which results in $R^2 = 0.987$, then follow XGBoost, random forest, and linear regression with R^2 metric 0.984, 0.952 and 0.920, respectively. As expected, the performance reduces with decreasing size of the training set. In Figure 4C the predictions of the reaction time by the XGBoost method based on 100 %, 34 %, and 10 % of the training set are depicted. The XGBoost model was chosen for this demonstration, because of its high performance. Moreover, the performance is less affected by a lowering of the size of the training set than the other two high performing models CatBoost and random forest considered in this investigation. The predictions based on only 34 % of the training set are still quite close to the given time by the kMC simulation. For 10 % of the training set the predictions are more scattered and the deviations of the predicted times from the kMC simulation results are too high. The observation for the prediction with the XGBoost model is underpinned by the results from Figure 4F and Table S1, showing the performance metrics R^2 of 0.927 and only 0.745 for a reduction of the training set to 34 % and 10 %, respectively. However, linear regression leads to R^2=0.916 even for training with the smallest training set of only 10 %. Thus, the recommendation is to use linear regression for extremely small data set sizes and more advanced ML methods, if enough training data is available. The explainability measure presented in Figure 4E indicates that the ranking of the influences of the input variables on the output is very similar for each model. The explainability of linear regression was excluded, because of the above-mentioned logarithmic transformation. The importance of the input variables $c_{\mathrm{BA},0}$ and T is almost equal with about 40 % and 35 %, respectively, followed by the importance of $c_{\mathrm{AIBN},0}$ with about 20 %.

6.3 Butyl acrylate concentration prediction

Equations (1), (3), and MTR from Equation (7) were used to predict c_{BA} and the distribution parameters M_{n} and M_{w} at different time points, keeping the dependency between dependent variable values for each of the considered models. A similar approach was considered in (Spyromitros-Xioufis et al., 2016), where product purchased after advertisement for each month throughout the year was predicted.

It is not necessary to predict c_{BA} values for each time moment, because the function to be predicted decreases monotonously. It is sufficient to predict the main points in the trajectory and approximate them (Figure 5).

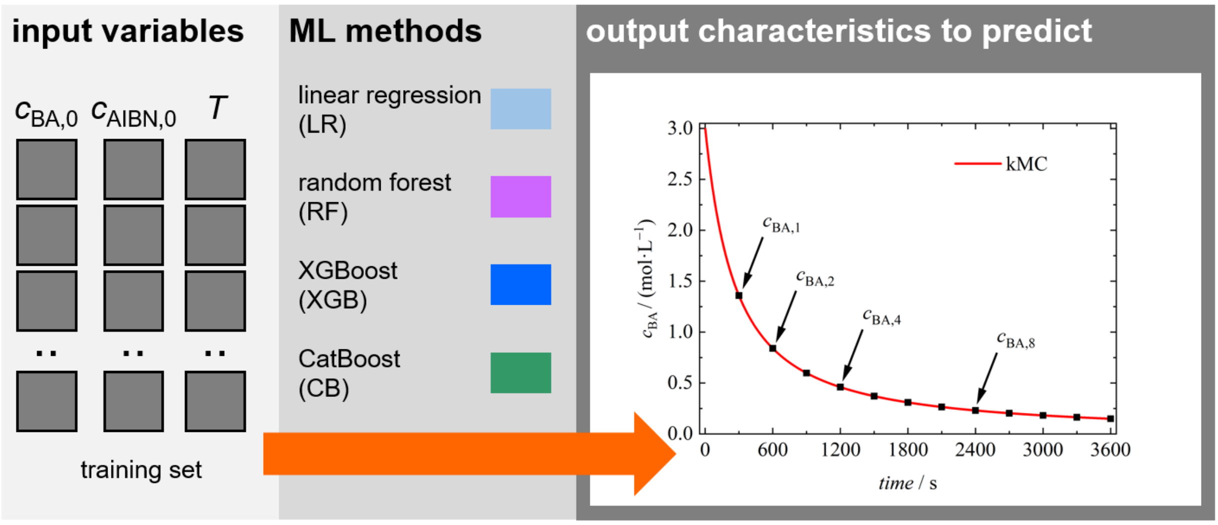

Figure 5: Monomer concentration prediction with multi-target regression. For further details see the main text.

The number of base time moments is denoted as d and the model should predict Y_1, Y_2, …, Y_d, which correspond to $c_{BA,i}$ at i time moments for each experiment. MTR was applied for keeping the dependence among the dependent variables, thus, $c_{BA,i}$ can depend on $c_{BA,i-1}$, $c_{BA,i-2}$, … at previous time moments in Equations (1). The same arguments regarding the MTR model are applicable to the prediction of the average molar masses M_n and M_w shown in Figure 3 and in Figure S1. Note, that AIBN concentration as a function of time $c_{AIBN}(t)$ is calculated directly with $c_{AIBN,0}$ according to the decomposition kinetics using Equation (8). Thus, no prediction of initiator concentration is required.

$$c_{AIBN}(t) = c_{AIBN,0} \cdot e^{-kt}, \tag{8}$$

where k is the rate coefficient for the AIBN decomposition reaction.

Figure 6 shows the predictions for the variation of c_{BA} with time obtained with the four above-described ML models. Figure 6A illustrates a typical prediction for one specific experiment. For visual comparison the simulated (ground truth) red line trajectory is compared with differently colored predicted trajectories of ML models. For this particular example, all the models provide good predictions. CatBoost performs best with an almost perfect overlap of the predicted and the kMC simulation-derived variation of c_{BA} with time. This finding is also generalized for the full training set by the performance metric R^2, whose results are shown in Figure 6B (for data see Table S3). Thus, the CatBoost method is associated with the best R^2 score of 0.999, followed by XGBoost, random forest and linear regression models with R^2 of

0.991, 0.981, and 0.845, respectively. It is found that the linear model performance is slightly weaker than for the other models, however, despite R^2 = 0.845 it is still considered to be very good. The more complex ensemble models random forest, XGBoost, and CatBoost yield R^2 = > 0.98 for the full training data set.

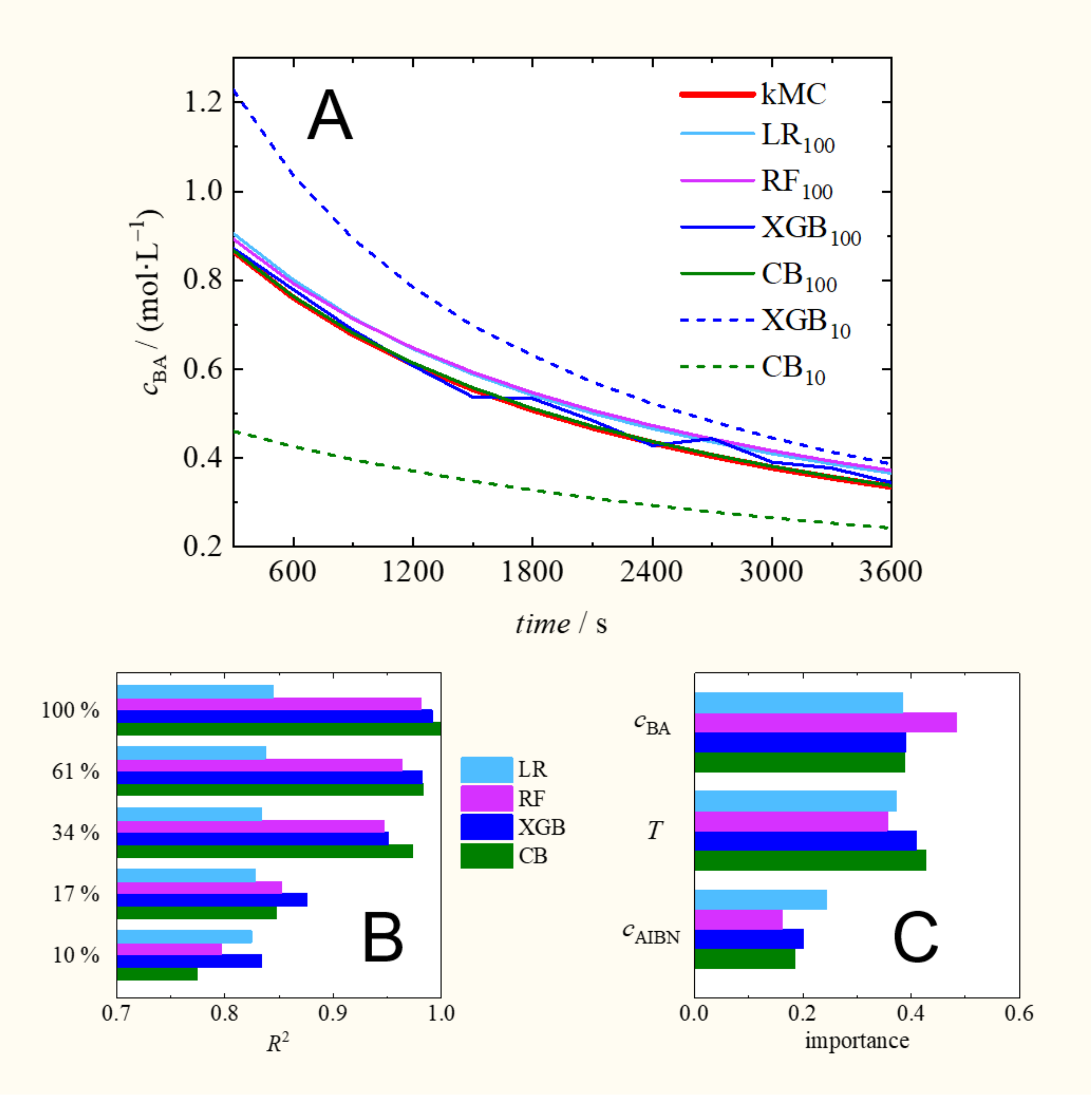

Figure 6: c_{BA} prediction models: (A) example of a prediction for T=70 °C, $c_{\mathrm{BA,0}}$=1.0 mol·L^{-1}, $c_{\mathrm{AIBN,0}}$=5 mmol·L^{-1}, training data set size: 432, (B) model performance, (C) feature importance.

The average performance of each model with different sizes of the training set is evaluated. As expected, the performance reduces with decreasing size of the training set as shown in Figure 6B. Remarkably, even with only 34 % of the training data the CatBoost model is associated with a performance metric value of $R^2 = 0.974$. The predictions with only 10 % of the training set even with the best performing methods CatBoost and XGBoost were not precise for the considered example (Figure 6A dashed lines). Thus, for the prediction of c_{BA}, the size of the training set can be significantly reduced. Investigations into the explainability of the considered models given in Figure 6C reveal that the ranking of the impact of the input variables on the output is very similar for each model. It is seen that $c_{\mathrm{BA,0}}$ is the most decisive input parameter for each model, accounting for around 40 % of the result. The importance of temperature is slightly lower followed by $c_{\mathrm{AIBN,0}}$ with a contribution of around 20 %. As expected, these results are similar to the explainability results from the reaction time prediction. The explainability results are meaningful from a chemical point of view, because the reaction time and the monomer concentration are directly related to the overall rate of polymerization. Moreover, the rate of polymerization is strongly affected by the radical concentration, which is given by the temperature and the initiator concentration. Therefore, a lowering of temperature can compensate for an increase of the initiator concentration and vice versa.

6.4 Prediction of average molar masses M_{n} and M_{w}

In Figure 7, the predictions for M_{w} with all methods considered are presented. Figure 7A contains an examplary prediction for a selected experiment. For this example, the line of CatBoost regression fully overlaps with the data from the kMC simulator. The linear regression results deviate significantly from the simulated data. Figure 7B and Table S4 contain the average performance of all the models estimated as R^2 metric on the full test set. All the models show R^2 above 0.917 for training with the full training set, e.g., the best-performing CatBoost model reaches a performance of 0.999. As expected the performance reduces with decreasing size of the training set for all models, except for the linear regression. Even with only 10 % of training data the linear regression model results in $R^2 = 0.90$, similar to the observation for the reaction time prediction. Thus, a significantly reduced training set is sufficient to predict M_{w}. This finding may be explained by the fact that unlike the other models the linear regression model does not require estimation of a big number of internal parameters. Therefore, its performance does not change considerably with major decrease in size of the training set. Explainability analysis depicted in Figure 7C indicates that again the ranking of the input variables' influence on the output is very similar for each model. The dominant feature is $c_{\mathrm{BA,0}}$

with about 70 %, followed by the impact of temperature and $c_{\mathrm{AIBN},0}$ with approximately 25 % and 5 %, respectively.

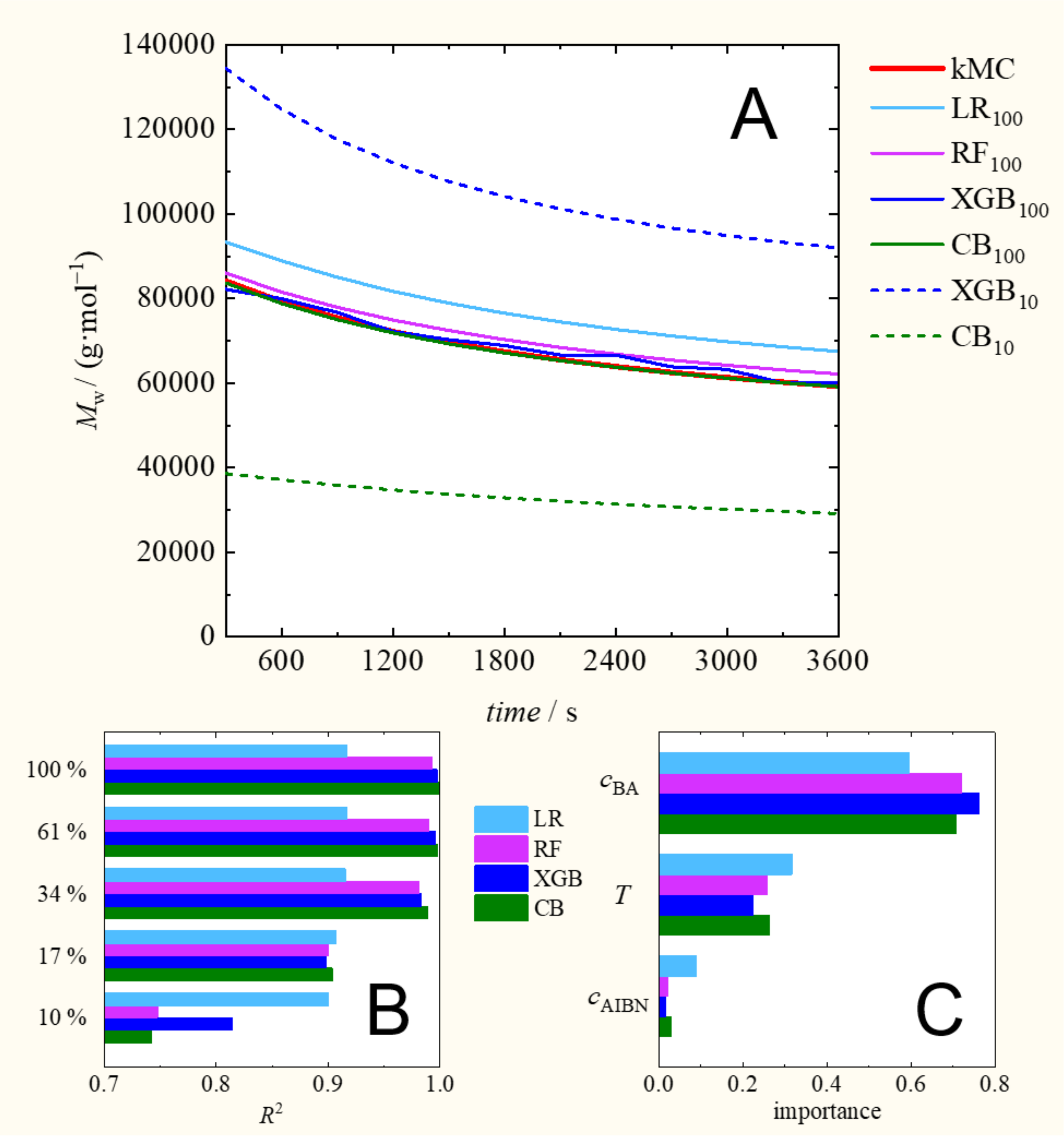

Figure 7: M_w prediction models: (A) example for a prediction with T=70 °C, $c_{\mathrm{BA},0}$=1.0 mol·L^{-1}, $c_{\mathrm{AIBN},0}$=5 mmol·L^{-1}, training data set size: 432 for all methods and strongly reduced training data set sizes of 10 % for XGBoost and CatBoost, (B) model performance for full and reduced training data sets for all methods, (C) feature importance for all methods and the full training data set.

The results for the prediction of M_n with various models are presented in Figure S2. As for M_w all the models provide good predictions for the considered example (Figure S2A). According to Figure S2B and Table S5 of the Supporting Information, again CatBoost and XGBoost are the best suited models and linear regression is the worst with R^2 = 0.881. The average performance of all models is above 0.881 taking R^2 as a metric. The CatBoost model performs best, indicated by R^2 reaching almost 1.0 for the full training set. As expected the performance reduces with decreasing size of the training set, however, even with 34 % of training data the CatBoost regression model gives R^2 = 0.984. In conclusion, as already observed for the predictability of M_w*,* for the prediction of M_n the training data set size can be significantly reduced. The results for the explainability of the considered models are similar as for the predictability of M_w. According to Figure S2C for all models considered the ranking of the variables' influence on the output is very similar: the importance of $\mathrm{c}_\mathrm{BA,0}$ is about 65 %, followed by temperature and $c_\mathrm{AIBN,0}$ with approximately 30 % and 5 %, respectively.

Since the training data set consists of in-silico generated data, which represents experimental data that are associated with experimental errors, it is interesting to check how the MMDs and molar mass averages predicted on the basis of different size training data sets deviate.

6.5 MMD prediction

The prediction of MMDs is more complex, because the MMD is a vector rather than a scalar value. For realistic forecasts the MMDs are described by 100 equidistant grid points. As the distribution represents a normalized weight fraction, the dependency between interval predictions can be realistically modeled with an MTR model. In this model, each output variable Y_i corresponds to one $w(\log (M))_i$ fraction interval, which is illustrated in Figure 8.

The mean squared error (MSE) was applied as a similarity metric for two curves, averaging the squared errors for each point of two MMDs for a given experiment.

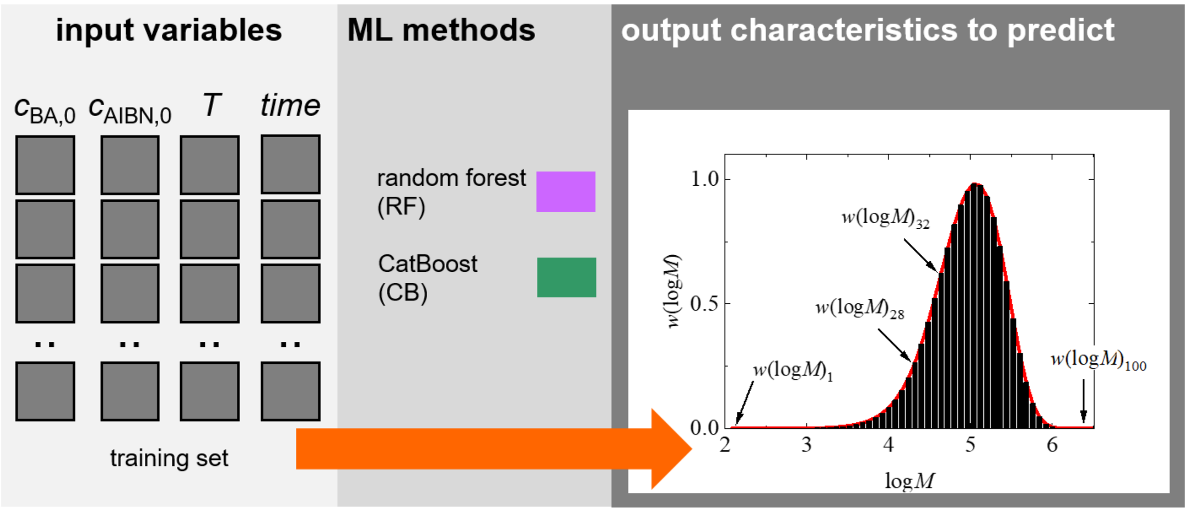

Figure 8: MMD prediction principles with multi-target regression. Further details are provided in the main text.

In Figure 9A MMDs are shown, which are predicted with the random forest and the CatBoost models according to Equation (4) for an example experiment. The corresponding MSEs are calculated for each MMD to simplify comparison, which is shown in Figure 9B and Table S6. Only random forest results are presented, because the other models failed. The linear regression model was too simple to describe such a complex task as the MMD prediction. It is supposed that the XGBoost model failed, because XGBoost supports an ensemble of single-target regressions instead of a traditional MTR. Moreover, the CatBoost method was unable to make predictions for less than 61 % of the full training set. In contrast, the random forest model provides a very good prediction of the MMDs. The average performance of random forest model is above 0.954 taking R^2 as a metric. The performance of CatBoost is slightly smaller with R^2=0.928 for the full training set.

As expected the performance reduces with decreasing size of the training set, however, the performance of the random forest model decreases less, even with 34 % of training data the random forest model yields R^2 = 0.927, while the CatBoost model is unable to make a prediction. For the random forest model, the size of the training set can be reduced to 34 % if required. We also proofed our findings by estimating the performance of the random forest model with a histogram of MSEs given in Figure S3, which shows that about 70 % of predictions for the last time moment of 3600 s are below $3 \cdot 10^{-4}$. Similar MSEs results were obtained for other time moments and model training with 61 % of the training data set. Upon further reduction of the training set, the MSEs increase proportionally with decreasing R^2 values.

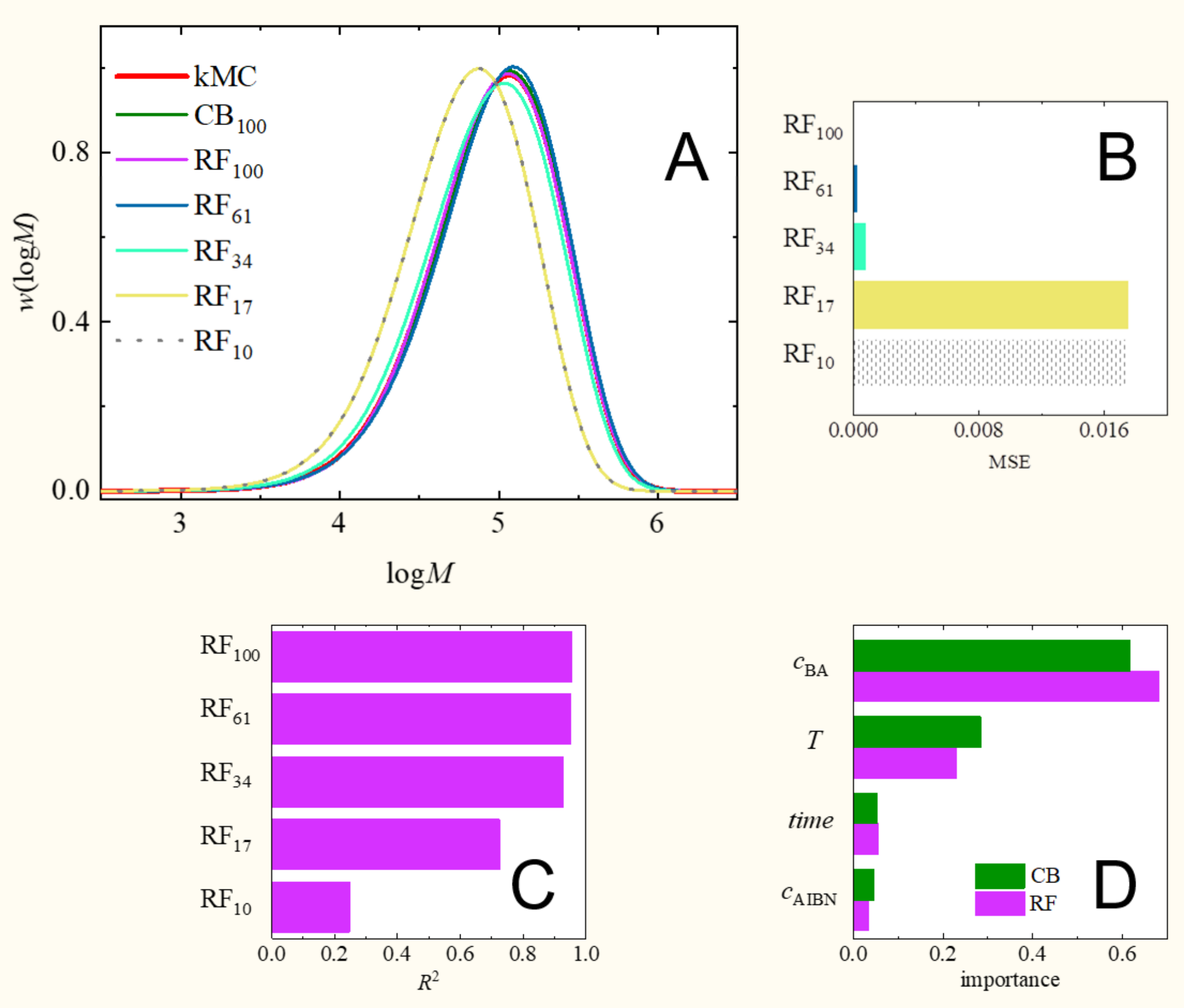

Figure 9: MMD prediction models: (A) example of a prediction for T=68 °C, $c_{BA,0}$=2.0 mol·L^{-1}, $c_{AIBN,0}$=10 mmol·L^{-1}, time=3600 s using the full training data set size of 432 for the CatBoost and random forest methods, as well as the indicated reduced training data set sizes for the random forest method. (B) MSEs of test set data of the models trained on reduced training sets, (C), model performance, (D) feature importances for the tests set.

In Figure 9D the model features are explained. In terms of explainability for both models the impact of $c_{BA,0}$ is dominant with a contribution higher than 60 %, followed by temperature with around 25 % as well as $c_{AIBN,0}$ and time with around 5 % each. Since M_n and M_w are derived from the MMD, the MMD explainability is similar to the explainability of the M_n and M_w predictions, which is meaningful from a chemical point of view. M_n is directly proportional to the rate of the propagation reaction, which is increased by monomer concentration. The temperature mainly affects the radical concentration due to its effect on the initiator decomposition reaction. Therefore, temperature and $c_{\mathrm{AIBN,0}}$ have a combined influence on the result and a change of one parameter can be compensated for by the other parameter.

6.6 Prediction of the polymerization conditions to yield a targeted MMD

While kMC simulations do not provide access to polymerization conditions leading to a targeted MMD, ML methods allow for building reverse engineering models in the same manner as polymerization process models, with the only difference that input and output variables are swapped. In this study, the first steps towards reverse engineering are reported. The single-objective ML models can serve as good starting points for the future multi-objective optimization procedure taking into account the criteria from section 3.3 (prediction scenarios) to identify the best polymerization conditions for a given MMD. There is no single solution to this problem, because different conditions can lead to the same MMD. However, in our case, $c_{\mathrm{BA,0}}$, $c_{\mathrm{AIBN,0}}$, and T for a given MMD are predicted with a single MTR model using Equation (6) and Equation (7). The results of the reverse engineering prediction are presented in Figure 10.

Firstly, in Figure 10 the reverse engineering approach is given for an exemplary prediction. In Figure 10A the kMC simulated MMD (red line) serves as input of the random forest reverse engineering model. Figures 10C, 10D, and 10E present the predictions of $c_{\mathrm{BA,0}}$, $c_{\mathrm{AIBN,0}}$ and T using the random forest model trained with different sizes of the training data set. Then, the parameters of the polymerization conditions predicted are used in a kMC simulation that provides the reverse engineering MMDs given for 100 %, 61 %, 34 %, 17 % and 10 % of the training set size. These MMDs show remarkable agreement with the input MMD (red) except for the smallest training set size, which is also illustrated by the MSEs for this example in Figure 10B. The MMD for 10 % of training set shows the biggest deviation, and consequently the highest MSE.

After considering an example forecast, we aim to discuss the performance of the ML model in general. In Figure 11B the model performance metric R^2 for the full and gradually reduced training sets is depicted. Employing the full training set the polymerization conditions are satisfactorily predicted according to $R^2 = 0.68$. The reason for the deviation observed may be due to the fact that the reaction time is fixed and conversions for both cases are expected to be different. Thus, the ML model provides only one potential polymerization condition, which is the optimal on the basis of minimizing the differences in conditions, taking only the MMDs and the corresponding conditions of the training set into account. Since multiple polymerization processes can lead to the same MMD, the next task is to propose a multi-objective optimization procedure, which takes monomer conversion and reaction time into account.

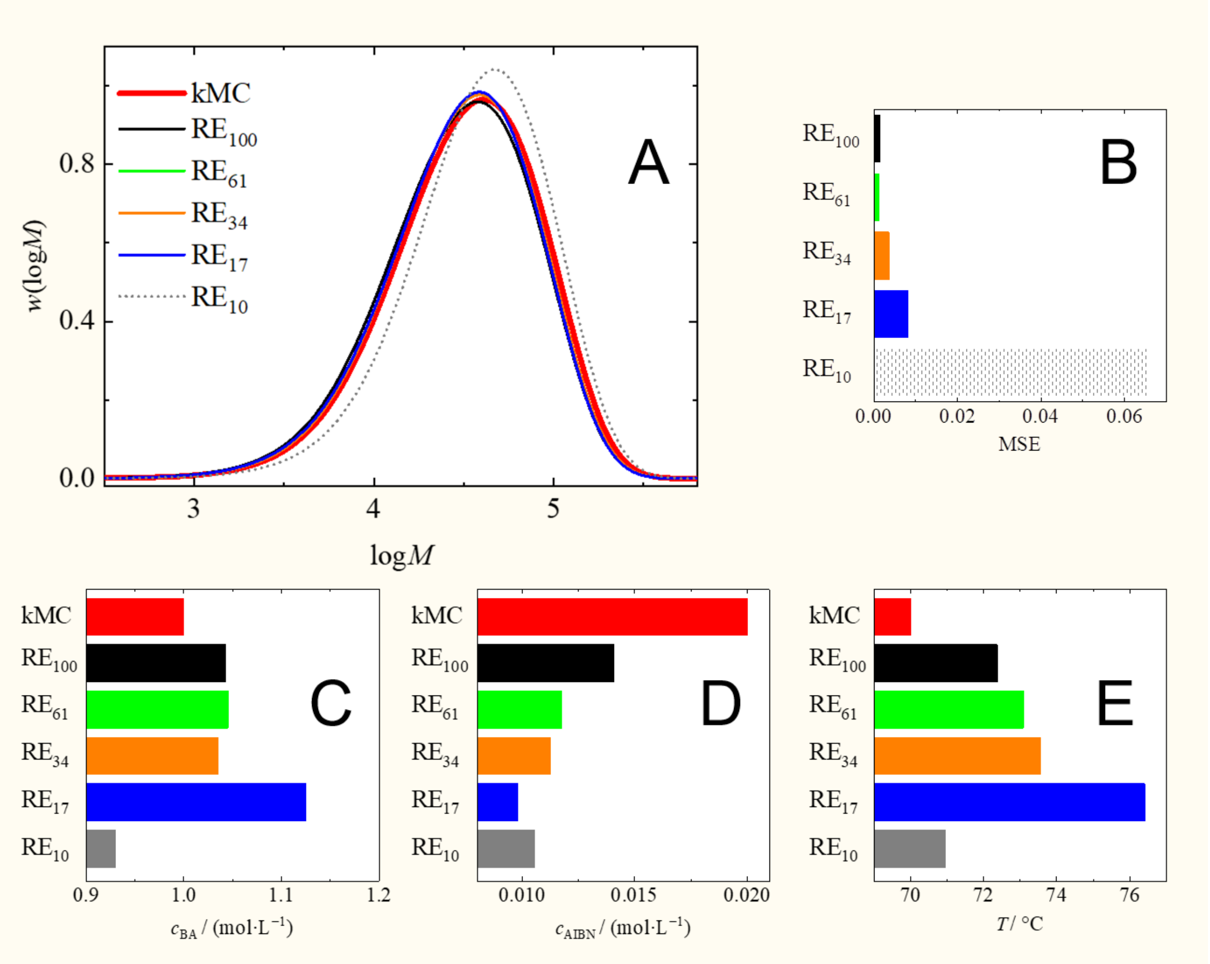

Figure 10: Prediction of the initial reaction mixture and polymerization temperature to obtain a pre-defined MMD (red): an example results for experiment: T=70 °C, $c_{\mathrm{BA,0}}$=1 mol·L^{-1}, $c_{\mathrm{AIBN,0}}$=20 mmol·L^{-1}, training set size: 432 with full and reduced training data sets, MMD prediction (A), random forest predicted recipes and temperature (C, D, E), MSEs (B).

As expected, the performance of the random forest model reduces with decreasing size of the training set, and for this case, the reduction of the training set size is not recommended. Figure 11A illustrates the MSEs calculated for each simulated experiment from the test set, which are increasing with the reduction of the training set size. Figure 11C allows for comparison of the average MSEs computed for the test set, for the random forest models fitted with gradually reduced training set. As expected, the MSEs of the random forest model fitted with 10 % of the training data are the largest, demonstrating the same tendency as for the shown example in Figure 10B.

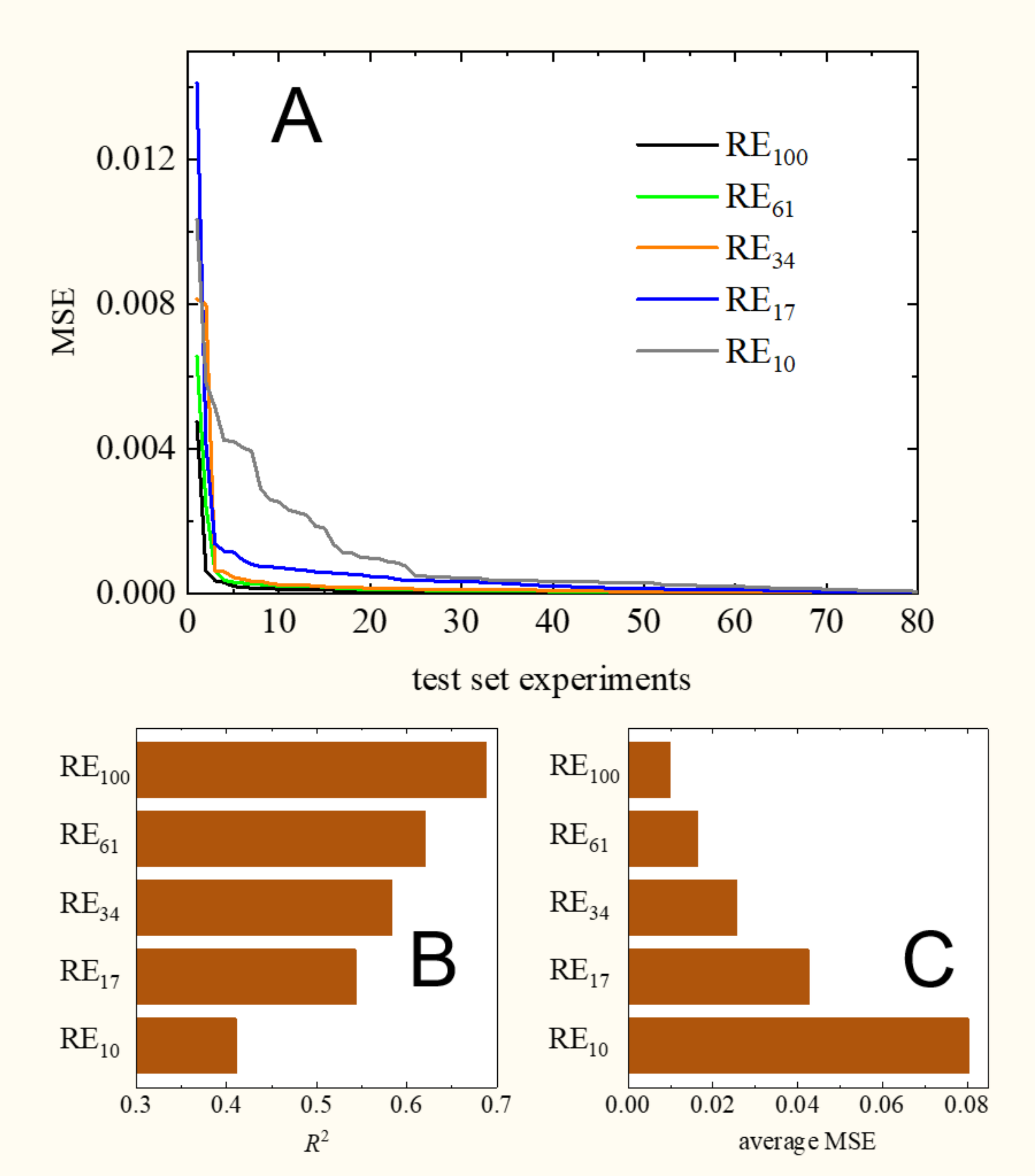

Figure 11: Random forest model prediction of the initial reaction mixture and polymerization temperature to obtain a pre-defined MMD: ordered MSEs for the test set data (A), total model performance (B), average MSEs for reduced training set data (C).

7 Conclusions

This study bridges the gap between state-of-the-art ML methods and their application in polymerization process modeling and reverse engineering of polymerization processes. The proposed polymerization process models allow for prediction of polymer molar masses and polymerization process conditions. The proposed reverse engineering models allow for simulation-supported prediction of a polymerization recipe and temperature for a targeted MMD.

A validated suite of scalable ML-based models for polymerization modeling was created, which facilitates fast and efficient simulation-supported learning of ML models. ML models for the prediction of four properties ($c_{\mathrm{BA},0}$, M_n, M_w, and MMD), reaction time and reverse engineering were proposed. The best models vary depending on the prediction task; however, usually decision-tree based ensemble models lead to a good result with $R^2 > 0.9$. Only the reverse engineering prediction model is associated with a lower performance of $R^2 = 0.68$. However, despite the low performance value very good agreement of the predicted and targeted MMDs is found. Moreover, almost all the models allow for reducing the training set size, without considerable loss in performance, which considerably decreases the number of kMC simulations necessary to train the ML models or even allows for training of the ML models with real-world laboratory experiments. In each of the considered 'black-box' models the feature importance was calculated, and the influence of each input variable was verified. From a polymer engineering point of view, the results are realistic and transparent. In future, the applied MTR and ML-based methodology will be generalized to predict other polymer and polymerization process properties (e.g., branching, polymerizations at high temperatures).

The first insights into the application of ML-based approaches to REPP were gained. Single-objective optimization was used, considering only the dependence between MMD and polymerization conditions as an optimization target. However, these insights and the created suite of ML-based models and methods allow us to solve more complex reverse engineering tasks in future, taking multiple objectives, e.g., as minimal reaction time, maximal conversion of monomer and initiator.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – <466601458> – within the Priority Programme "SPP 2331: Machine Learning in Chemical Engineering".

Appendix A. Supplementary data

Supplementary material related to this article can be found online at: (attached to this submission).

References

Afanasyeva, H. (2002). Fuzzy Learning Classifiers Systems for Classification Task. *Transport and Telecommunication*, *3*(3), 43–51. Retrieved from https://pdfs.semanticscholar.org/43b6/20608cacfddfdc3084de85ec4073189b5f90.pdf

Ancona, M., Ceolini, E., Öztireli, C., & Gross, M. (2018). Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *In Proc. of 6th Int. Conf. on Learning Representations, ICLR* .

Borchani, H., Varando, G., Bielza, C., & Larrañaga, P. (2015). A survey on multi-output regression. *WIREs Data Mining and Knowledge Discovery*, *5*(5), 216–233. doi:10.1002/widm.1157

Brandl, F., Drache, M., & Beuermann, S. (2018). Kinetic Monte Carlo Simulation Based Detailed Understanding of the Transfer Processes in Semi-Batch Iodine Transfer Emulsion Polymerizations of Vinylidene Fluoride. *Polymers*, *10*(9). doi:10.3390/polym10091008

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification And Regression Trees*: Routledge.

Castro, J., Gómez, D., & Tejada, J. (2009). Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research*, *36*(5), 1726–1730. doi:10.1016/j.cor.2008.04.004

CatBoost library. (2023). Retrieved from https://catboost.ai

Charoenpanich, T., Anantawaraskul, S., & Soares, J. B. P. (2020). Using Artificial Intelligence Techniques to Design Ethylene/1-Olefin Copolymers. *Macromolecular Theory and Simulations*, *29*(6), 2000048. doi:10.1002/mats.202000048

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In : *KDD '16, Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (pp. 785–794). New York, NY, USA: Association for Computing Machinery.

Curteanu, S. (2004). Direct and inverse neural network modeling in free radical polymerization. *Open Chemistry*, *2*(1), 113–140. doi:10.2478/BF02476187

Curteanu, S., Leon, F., Mircea-Vicoveanu, A.-M., & Logofătu, D. (2021). Regression Methods Based on Nearest Neighbors with Adaptive Distance Metrics Applied to a Polymerization Process. *Mathematics*, *9*(5), 547. doi:10.3390/math9050547

Da Tan, J., Ramalingam, B., Wong, S. L., Cheng, J., Lim, Y.-F., Chellappan, V., …Hippalgaonkar, K. (2022). Machine Learning Predicts Conversion and Molecular Weight Distributions in Computer Controlled Polymerization. *ChemRxiv*. doi:10.26434/chemrxiv-2022-tlz53

Dall Agnol, L., Ornaghi, H. L., Monticeli, F., Dias, F. T. G., & Bianchi, O. (2021). Polyurethanes synthetized with polyols of distinct molar masses: Use of the artificial neural network for prediction of degree of polymerization. *Polymer Engineering & Science*, *61*(6), 1810–1818. doi:10.1002/pen.25702

D'hooge, D. R. (2018). In Silico Tracking of Individual Species Accelerating Progress in Macromolecular Engineering and Design. *Macromolecular rapid communications*, *39*(14), e1800057. doi:10.1002/marc.201800057

Drache, M. (2009). Modeling the Product Composition During Controlled Radical Polymerizations with Mono- and Bifunctional Alkoxyamines. *Macromolecular Symposia*, *275–276*(1), 52–58. doi:10.1002/masy.200950106

Drache, M., & Drache, G. (2012). Simulating Controlled Radical Polymerizations with mcPolymer—A Monte Carlo Approach. *Polymers*, *4*(3), 1416–1442. doi:10.3390/polym4031416

Drache, M., Hosemann, B., Laba, T., & Beuermann, S. (2015). Modeling of Branching Distributions in Butyl Acrylate Polymerization Applying Monte Carlo Methods. *Macromolecular Theory and Simulations*, *24*. doi:10.1002/mats.201400081

Dragoi, E., & Curteanu, S. (2016). The use of differential evolution algorithm for solving chemical engineering problems. *Reviews in Chemical Engineering*, *32.* doi:10.1515/revce-2015-0042

Draper, N. R., & Smith, H. (1998). *Applied Regression Analysis*. New York: Wiley.

Edeleva, M., Marien, Y. W., van Steenberge, P. H. M., & D'hooge, D. R. (2021). Impact of side reactions on molar mass distribution, unsaturation level and branching density in solution free radical polymerization of n -butyl acrylate under well-defined lab-scale reactor conditions. *Polymer Chemistry*, *12*(14), 2095–2114. doi:10.1039/D1PY00151E

Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Ecology*, *77*(4), 802–813.

Fernandes, F., & Lona, L. (2005). Neural network applications in polymerization processes. *Brazilian Journal of Chemical Engineering - BRAZ J CHEM ENG*, *22.* doi:10.1590/S0104-66322005000300009

Feuerpfeil, A., Drache, M., Jantke, L.-A., Melchin, T., Rodríguez-Fernández, J., & Beuermann, S. (2021). Modeling Semi-Batch Vinyl Acetate Polymerization Processes. *Industrial & Engineering Chemistry Research*, *60*(50), 18256–18267. doi:10.1021/acs.iecr.1c03114

Fiosina, J., Fiosins, M., & Müller, J. P. (2013). Mining the Traffic Cloud: Data Analysis and Optimization Strategies for Cloud-Based Cooperative Mobility Management. In J. Casillas, F. J. Martínez-López, R. Vicari, & F. de La Prieta (Eds.), *Advances in Intelligent Systems and Computing. Management Intelligent Systems* (pp. 25–32). Springer.

Ghiba, L., Drăgoi, E. N., & Curteanu, S. (2021). Neural network-based hybrid models developed for free radical polymerization of styrene. *Polymer Engineering & Science*, *61*(3), 716–730. doi:10.1002/pen.25611

Ghosn, J., & Bengio, Y. (1996). Multi-Task Learning for Stock Selection. In : *NIPS'96, Proc. of the 9th Int. Conf. on Neural Information Processing Systems* (pp. 946–952). Cambridge, MA, USA: MIT Press.

Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, *22*(4), 403–434. doi:10.1016/0021-9991(76)90041-3

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The elements of statistical Learning*: Springer.

Hernández-Ortiz, J. C., van Steenberge, P. H. M., Reyniers, M.-F., Marin, G. B., D'hooge, D. R., Duchateau, J. N. E., …Schreurs, F. (2017). Modeling the reaction event history and microstructure of individual macrospecies in postpolymerization modification. *AIChE Journal*, *63*(11), 4944–4961. doi:10.1002/aic.15842

Holzinger, A. (2018). *Explainable AI (ex-AI)*. Retrieved from https://gi.de/informatiklexikon/explainable-ai-ex-ai/

Iedema, P. D., & Hoefsloot, H. C. J. (2006). Conditional Monte Carlo Sampling To Find Branching Architectures of Polymers from Radical Polymerizations with Transfer to Polymer. *Macromolecules*, *39*(8), 3081–3088. doi:10.1021/ma052535o

Jhaveri, S., Khedkar, I., Kantharia, Y., & Jaswal, S. (2019). Success Prediction using Random Forest, CatBoost, XGBoost and AdaBoost for Kickstarter Campaigns. In *Proc. 3rd Int. Conf. on Computing Methodologies and Communication (ICCMC)* (pp. 1170–1173).

Karkera, K. (2017). Regression Models with multiple target variables. *Towards Data Science*,

Kocev, D., Džeroski, S., White, M., Newell, G., & Griffioen, P. (2009). Using single- and multi-target regression trees and ensembles to model a compound index of vegetation condition. *Ecological Modelling - ECOL MODEL*, *220*, 1159–1168. doi:10.1016/j.ecolmodel.2009.01.037

Li, H., Collins, C. R., Ribelli, T. G., Matyjaszewski, K., Gordon, G. J., Kowalewski, T., & Yaron, D. J. (2018). Tuning the molecular weight distribution from atom transfer radical polymerization using deep reinforcement learning. *Mol. Syst. Des. Eng.*, *3*(3), 496–508. doi:10.1039/C7ME00131B

Liu, Z. P., Courteanu, S., Fischer, A., Jelfs, K., Oganov, A. R., Laino, T., …others. (2020). *Machine Learning in Chemistry: The Impact of Artificial Intelligence*. *Issn Series*: Royal Society of Chemistry. Retrieved from https://books.google.de/books?id=g7PAywEACAAJ

Lundberg, S. M., & Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 4765–4774). Curran Associates, Inc.

Martin, T. B., & Audus, D. J. (2023). Emerging Trends in Machine Learning: A Polymer Perspective. *ACS Polymers Au.* doi:10.1021/acspolymersau.2c00053

Meyer, A. (2021). Multi-target normal behaviour models for wind farm condition monitoring. *Applied Energy, 300*, 117342. doi:10.1016/j.apenergy.2021.117342

Mohammadi, Y., & Penlidis, A. (2018). Polymerization Data Mining: A Perspective. *Advanced Theory and Simulations, 2.* doi:10.1002/adts.201800144

Mohammadi, Y., Saeb, M. R., Penlidis, A., Jabbari, E., J. Stadler, F., Zinck, P., & Matyjaszewski, K. (2019). Intelligent Machine Learning: Tailor-Making Macromolecules. *Polymers, 11*(4). doi:10.3390/polym11040579

Molnar, C. (2022). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable* (2nd ed.). Retrieved from https://christophm.github.io/interpretable-ml-book

MongoDB: The Developer Data Platform. (2023). Retrieved from https://www.mongodb.com/

Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. In *Proc. of the IEEE Conf. on computer vision and pattern recognition* (pp. 427–436).

Peikert, P., Pflug, K. M., & Busch, M. (2019). Modeling of High-Pressure Ethene Homo- and Copolymerization. *Chemie Ingenieur Technik, 91*(5), 673–677. doi:10.1002/cite.201800206

Prokhorenkova, L. O., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *NeurIPS* (pp. 6639–6649).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why Should I Trust You?": Explaining the Predictions of Any Classifier. In : *KDD '16, Proc. of 22Nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (pp. 1135–1144). New York, NY, USA: ACM.

Rokach, L. (2019). *Ensemble Learning* (2nd): World Scientific.

Saldívar-Guerra, E. (2020). Macromol. React. Eng. 4/2020. *Macromolecular Reaction Engineering, 14*(4), 2070007. doi:10.1002/mren.202070007

Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning Important Features Through Propagating Activation Differences. In *Proc. of 34th Int. Conf. on ML, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017* (pp. 3145–3153).

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., & Vlahavas, I. (2016). Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning, 104.* doi:10.1007/s10994-016-5546-z

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic Attribution for Deep Networks. In : *ICML'17, Proc. of the 34th Int. Conf. on ML* (pp. 3319–3328). JMLR.org.

Trigilio, A. D., Marien, Y. W., Edeleva, M., van Steenberge, P., & D'hooge, D. R. (2022). Optimal search methods for selecting distributed species in Gillespie-based kinetic Monte Carlo. *Computers & Chemical Engineering, 158,* 107580. doi:10.1016/j.compchemeng.2021.107580

Trigilio, A. D., Marien, Y. W., van Steenberge, P. H. M., & D'hooge, D. R. (2020). Gillespie-Driven kinetic Monte Carlo Algorithms to Model Events for Bulk or Solution (Bio)Chemical Systems Containing Elemental and Distributed Species. *Industrial & Engineering Chemistry Research, 59*(41), 18357–18386. doi:10.1021/acs.iecr.0c03888

van Steenberge, P. H. M., Vandenbergh, J., Reyniers, M.-F., Junkers, T., D'hooge, D. R., & Marin, G. B. (2017). Kinetic Monte Carlo Generation of Complete Electron Spray Ionization Mass Spectra for Acrylate Macromonomer Synthesis. *Macromolecules, 50*(7), 2625–2636. doi:10.1021/acs.macromol.7b00333

Zhang, J., & Pantelelis, N. (2011). Modelling and Control of Reactive Polymer Composite Moulding Using Bootstrap Aggregated Neural Network Models. *Chemical Product and Process Modeling, 6.* doi:10.2202/1934-2659.1603