On the use of Machine Learning modeling tools in chemical process and product engineering

Cindy Trinh

UNIVERSITÉ DE LORRAINE

École doctorale SIMPPÉ

On the use of Machine Learning modeling tools in chemical process and product engineering

Doctoral Dissertation
(CONFIDENTIAL)

submitted and defended publicly on November 24^{th} 2023

in a partial fulfillment of the requirements for obtaining a

PhD title from the University of Lorraine

Specialty : Process, Product and Molecular Engineering

by

Cindy TRINH

Composition of the jury :

President :	**Bernardetta ADDIS**, Professor	LORIA, Nancy
Reviewers :	**Ludovic MONTASTRUC**, Professor	INP ENSIACET, Toulouse
	Florence VERMEIRE, Professor	KU Leuven, Leuven
Examiners :	**Amanda L.T. BRANDÃO**, Professor	PUC Rio, Rio de Janeiro
Guests :	**Silvia LASALA**, Lecturer	Université de Lorraine, Nancy
	Thibaut NEVEUX, PhD engineer	EDF R&D, Chatou
Supervisors :	**Dimitrios MEIMAROGLOU**, Lecturer	Université de Lorraine, Nancy
	Sandrine HOPPE, Research fellow	Université de Lorraine, Nancy

Laboratoire Réactions et Génie des Procédés — UMR 7274

Mis en page avec la classe thesul.

Acknowledgments

These three years of thesis have been a very positive experience to me, and I would like to thank all the people who contributed in making it possible, in one way or another.

First of all, I would like to express my profound gratitude to my supervisor, Dimitrios Meimaroglou, for having trusted me from the very beginning. It has been a real pleasure to work under your positive guidance: your advice and encouragements motivated me a lot and made me feel very inspired in my work. I especially appreciated your kindness, willing to help and deep implication in reviewing my work. With this experience, I feel that I finally found what I really like to do, and for that, I am very grateful.

I also would like to address my sincere gratitude to my co-supervisor, Sandrine Hoppe, also for having gave me the chance to do this doctoral project. Thank you for your caring advice and encouragements and for providing me all the necessary materials, chemicals and support to implement the experiments.

This research was funded by MESRI (Ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation), and by the Institute Carnot ICEEL (Grant: "Recyclage de Pneus par Intelligence Artificielle - RePnIA"), France. Therefore I would like to thank them for funding my research.

I also would like to sincerely thank Florence Vermeire and Ludovic Montastruc for attentively reviewing this work and for their constructive remarks. I would like to thank all the members of my defense committee as well, for having accepted to be part of it and for their great interest in my work. It has been very enriching to discuss with you during the defense and I would like to thank you all very much for your interesting questions and advice to improve my work.

For some parts of this thesis, I received a valuable help from other colleagues of LRGP/ERPI and I would like to thank them greatly. I would like to thank Richard Lainé and the team of Charly Koenig for their great reactivity and technical support to face any challenging experimental issue. I also would like to thank Silvia Lasala and Olivier Herbinet for their kindness, for their great interest in my work and their critical feedback to improve it (and also for the pleasant and delicious dinners). To the members of my follow-up committee, Jean-Marc Commenge and Mauricio Camargo, I also would like to express my thank for their time and constructive remarks. Thank you to my different interns (Betty, Bahia, Jérémie, Justine, Youssef, Redouane and Wenbo) for having supported me so greatly in the experimental and modeling parts.

Obviously, I would like to warmly thank my greedy colleagues/friends from the ground floor for their good mood, their support and the always interesting non-scientific discussions and eating international food sessions. Thank you so much Pan(da), Alejandra, Pilar (and Jérémie), Tiantian and of course Véronique. I was so happy to share the same office with the best colleagues ever and I will miss you all very much.

Evidently, I do not forget my other colleagues from the other floors/buildings, my friends since ENSIC, since the computer project: Dr Don Pietro and Dr Fátima. Thank you so much for your loyal friendship, I was really happy that we could share again many convivial moments in Nancy. Of course, there are also many people that I would like to thank for encouraging me all along the thesis: Hao, Manon, Kim, Ulysse, Pauline, Françoise, Mosbah, Mengxi, Yanying, Alexis, Cristian and many others.

Last but not least, I would like to express my deepest thanks to my loved ones: my parents, my sister & Martial and my favorite boys Benjamin (=the 1st favorite ;)) & Nori. Thank you so much for your never-ending devotion, affection and support. I feel so lucky for having you all and I will always feel so grateful for all the love (or purring) you give me every single day ♡.

Contents

General introduction

General context

Artificial intelligence (AI), machine learning (ML) and deep learning (DL) methods have gained growing interest in recent decades, and the Chemical Product Engineering (CPE) sector is not exempt. Indeed, these data-driven methods, which have developed greatly thanks to the explosion of data availability and to technological/mathematical advances, have succeeded in solving complex problems in other fields (e.g., image generation, autonomous driving cars, natural language processing). As a result, they could be very interesting for certain CPE challenges, when classical approaches reach their limits. For example, phenomenological methods (i.e., based on the description of the mechanisms and the phenomena that drive a process), when they exist, do not always manage to correctly capture the relationship between the manufacturing process, structures, ingredients and usage properties of complex CPE products. Indeed, the latter are generally characterized by their multiple structures, end-use properties and/or ingredients. Another example concerns the design and discovery of new molecules or materials with given properties, which necessitates new methods for a better exploration of the vast chemical space. At the same time, the ML field contains a plethora of methods and concepts that are sometimes difficult or time-consuming to master for newcomer chemical engineers willing to apply these approaches. Additionally, the success of ML methods should not render their application systematic in any problem as these methods also possess their own requirements and limits in comparison with knowledge-based approaches.

Motivations and objectives

This thesis focuses on the use of ML approaches in the context of CPE.

In the first part, the aim is to provide a **state-of-the-art** and a better understanding of the characteristics, advantages, and drawbacks of these approaches and how they apply depending on the nature of CPE applications (e.g. often limited amount of data, complex representation of molecules, reactions, spectra or sensory properties). The main challenges of ML in CPE are also discussed and some guidelines are provided for the selection of a ML method.

In the second part, the goal is to test different ML approaches on two concrete CPE applications, where classical approaches reach their limits. These are characterized by very different objectives and data characteristics, thus impacting the adopted approaches.

The **first application** concerns the development of a QSPR (quantitative structure-property relationship) model to predict two thermodynamic properties of molecules from their structural and physico-chemical features (a.k.a. descriptors). In particular, the considered data set is characterized by two challenging aspects. The first is the high-dimensional description of the molecules to increase the chances to capture the relevant features affecting the thermodynamic properties, in absence of knowledge. The second is the wide diversity of chemical structures in order to enlarge the applicability of the developed model for future chemical discovery purposes. Above all, the special feature of this application lies within the multi-angle approach adopted to better visualize and understand the possible methods at each stage (from data collection to model construction) and their impact on the model obtained. Additionally, methods to define the applicability domain and to detect the outliers in this high-dimensional problem are investigated.

The **second application** deals with the modeling of a styrene polymerization process in the presence (and in the absence) of used tire particles (a.k.a. ground tire rubber, GTR), in order to predict the styrene conversion rate as a function of the prevailing operating conditions. In this case, the data set is limited (as generated by time-consuming experiments) but the mechanisms and kinetics of some parts of the system are well-known. In this sense, both pure ML methods and hybrid ML/knowledge-based model methods are developed and compared, with a particular interest in Gaussian processes for the ML part, to provide uncertainty in predictions. As for the knowledge-based part, the kinetic model developed during a precedent thesis is exploited.

Outline

This manuscript is divided into 5 chapters as presented below. Chapter 1 corresponds to the state-of-the-art, while the other chapters present the applications (Chapters 2 and 3 for the first application and Chapters 4 and 5 for the second application). Note that this manuscript is publication-based, meaning that the content of the presented chapters is heavily based on a series of scientific articles that have been prepared during the thesis and reflect the totality of the work carried out. Some of them are already submitted while others are, at the moment that this text is prepared, under preparation for submission. Note that articles presented in their published version are open access articles, covered by the Creative Commons Attribution (CC BY) license.

- **Chapter 1** *(published)*: State-of-the-art of Machine Learning in chemical product engineering
- **Chapter 2** *(published)*: A comprehensive methodology for the development of descriptor-based QSPR/QSAR models
- **Chapter 3** *(published)*: Descriptor-based QSPR/QSAR models: the applicability domain problem in high dimension
- **Chapter 4** *(published)*: Machine Learning modeling of the styrene–GTR radical graft polymerization
- **Chapter 5** *(for future publication)*: Hybrid modeling and Gaussian Processes in polymer engineering

IMPORTANT NOTE: The present manuscript was submitted on September 27th, 2023 under a confidentiality request for a period of 6 months starting from this date. A part of the work is indeed related to a project that is currently the subject of a patent application procedure.

Chapter 1

State-of-the-art of Machine Learning in chemical product engineering

Contents

1.1 Introduction

In view of the abundance of ML methods and their applications in CPE, this first chapter provides an introduction to ML, while giving an overview of the use of ML in the field of CPE when classical approaches reach their limits. The characteristics and principles of different ML methods are explained and a special emphasis is given on the following CPE applications:

- design and discovery of new molecules and materials
- process modeling
- support for sensory analysis
- prediction of chemical reactions

These applications were chosen to exploit the different parts of the well-known "process-structure-property" relationship of CPE systems. For example, the first one investigates the relationship between structure and property. The second one focuses mainly on the prediction of property based on process parameters. The third one studies the impact of process conditions, ingredients and/or structural characteristics on sensory properties. The last one is more related to organic chemistry, however the prediction of reactions (products, reactants, experimental conditions) is crucial for example to synthesize the chemicals present in the three aforementioned applications.

More generally, all these applications display their own data characteristics and challenges (e.g. limited amount of data, complex representation of molecules, reactions, spectra or sensory properties) and therefore the adopted ML approaches can vary significantly.

Even if ML methods display successful results to solve different CPE challenges, it is however important to highlight that the resort to ML methods is not without risks. These approaches possess specific advantages and drawbacks, which one should be aware of before starting any ML application.

Despite the great number of reviews in the field, there is a lack of a synthetic publication addressing the specific challenges of ML-CPE and providing a guide to newcomers with the major pitfalls and points of interest. This is what is proposed in the present chapter.

1.2 Outline of the publication

1. **Introduction**
2. **ML: Background**
   - 2.1 Categories of ML Algorithms
     - Supervised learning
     - Unsupervised learning
     - Semi-supervised learning
     - Reinforcement learning
   - 2.2 Hybrid and Combinatorial Approaches
3. **ML in Chemical Product Engineering: State-of-the-Art**
   - 3.1 Current Challenges in Chemical Product Engineering and Role of AI/ML
   - 3.2 Overview of ML Methods in Chemical Product Engineering
     - Design and discovery of new molecules and materials
     - Prediction of chemical reactions and retrosynthesis
     - Modeling and optimization of process–properties relationship
     - Support for sensorial analysis
4. **Guidelines for Applying ML in Chemical Product Engineering Problems**
   - 4.1 General Principle of Some Popular ML Methods in Chemical Product Engineering
     - ANN
     - SVM
     - GP
     - PCA
     - Other ML methods
   - 4.2 Interest of Data-Driven Methods
   - 4.3 Challenges and Solutions
     - Data
     - Lack of understanding
   - 4.4 General Guidelines for the Selection of a ML Method
5. **Conclusions**

1.3 Publication *"Machine Learning in Chemical Product Engineering: The State-of-the-Art and a Guide for Newcomers", published on 20 August 2021*

processes

MDPI

Review

Machine Learning in Chemical Product Engineering: The State of the Art and a Guide for Newcomers

Cindy Trinh , Dimitrios Meimaroglou * and Sandrine Hoppe

Laboratoire Réactions et Génie des Procédés, Université de Lorraine, CNRS UMR7274, LRGP, F-54000 Nancy, France; cindy.trinh@univ-lorraine.fr (C.T.); sandrine.hoppe@univ-lorraine.fr (S.H.)
* Correspondence: dimitrios.meimaroglou@univ-lorraine.fr

Abstract: Chemical Product Engineering (CPE) is marked by numerous challenges, such as the complexity of the properties–structure–ingredients–process relationship of the different products and the necessity to discover and develop constantly and quickly new molecules and materials with tailor-made properties. In recent years, artificial intelligence (AI) and machine learning (ML) methods have gained increasing attention due to their performance in tackling particularly complex problems in various areas, such as computer vision and natural language processing. As such, they present a specific interest in addressing the complex challenges of CPE. This article provides an updated review of the state of the art regarding the implementation of ML techniques in different types of CPE problems with a particular focus on four specific domains, namely the design and discovery of new molecules and materials, the modeling of processes, the prediction of chemical reactions/retrosynthesis and the support for sensorial analysis. This review is further completed by general guidelines for the selection of an appropriate ML technique given the characteristics of each problem and by a critical discussion of several key issues associated with the development of ML modeling approaches. Accordingly, this paper may serve both the experienced researcher in the field as well as the newcomer.

Keywords: machine learning; artificial intelligence; chemical product engineering; data-driven modeling; materials design; sensorial analysis; prediction of chemical reactions

Citation: Trinh, C.; Meimaroglou, D.; Hoppe, S. Machine Learning in Chemical Product Engineering: The State of the Art and a Guide for Newcomers. *Processes* **2021**, *9*, 1456. https://doi.org/10.3390/pr9081456

Academic Editor: Andrew Hoadley

Received: 1 July 2021
Accepted: 4 August 2021
Published: 20 August 2021

1. Introduction

Artificial intelligence (AI) and machine learning (ML) have gained increasing interest among chemical and process engineers over the last decade. AI can be defined as a set of methods enabling to reproduce human behavior in order to solve high complexity problems, such as speech recognition, linguistic translation and image analysis. ML is a subset of AI, referring to a set of algorithms whose performance, relative to a given task, improves upon receiving more and more relevant data (i.e., the computer program is considered to be learning from experience) [1]. Given the dataset the user will provide to the algorithm, the latter will identify on its own, without being explicitly programmed by the user, eventual mathematical correlations and patterns among them.

This current great popularity of AI and ML is mostly driven by the increasingly facilitated access to large amounts of data of diverse variety along with the major advances in modern computational systems that are becoming more powerful and affordable every day. This rapid evolution is illustrated in Figure 1, where the number of annually published documents (including articles, proceedings papers, reviews and book chapters), containing AI- and ML-related keywords in their title, are plotted for all types of applications and for chemistry-related applications (i.e., materials science, chemical engineering, biochemistry etc.), on the left- and right-hand sides, respectively.

In addition, ML methods have already shown promising potential in tackling complex problems in various fields (e.g., robotics, computer vision and natural language processing), as well as in chemical engineering and Chemical Product Engineering (CPE), such as the

discovery of new molecules with targeted functional properties or the optimization of process conditions to obtain specific properties.

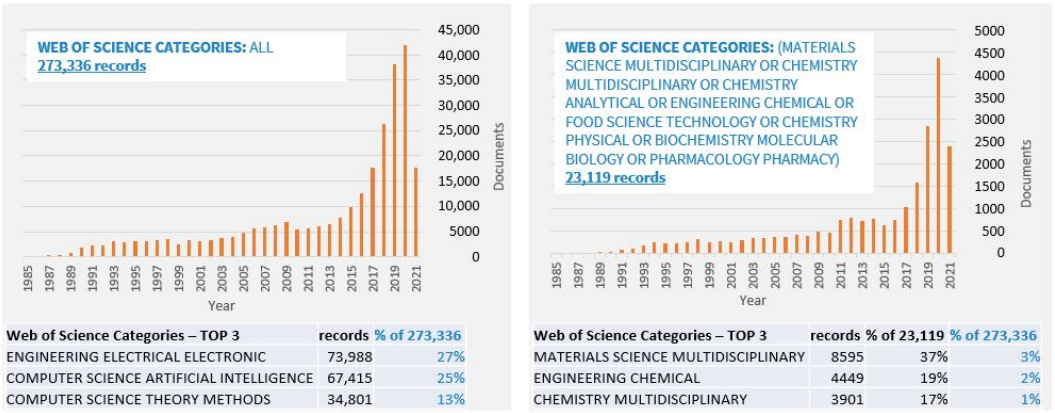


Figure 1. Evolution of the number of annually published documents (including articles, proceedings papers, reviews and book chapters), containing the following keywords in their title: "machine learning" or "artificial intelligence" or "AI" or "deep learning" or "data driven" or "neural network". (**left**) All categories of Web of Science are included. (**right**) Only categories related to chemistry are included.

CPE refers to the field of science that studies the different processes and methodological approaches aiming at elaborating products or materials of specifically identified tailor-made properties and functionalities. In particular, these products are characterized by strong interactions between process parameters, ingredient characteristics (e.g., composition, properties…) and final product properties and structure. There are numerous challenges associated with the modeling of these products and systems, mostly related to their multi-parametric, complex nature. Indeed, products like cosmetics or emulsions, are most often multifunctional and/or multi-ingredient and present a specific need in controlling several end-use characteristics and properties.

For example, paints must display a specific range of aesthetic, resistance and rheological functions, in order to respond to the various constraints related to their transport, storage, application and longevity demands. In addition, the understanding of the link between process, ingredients and product structure and properties is not a trivial task to accomplish, given the increased associated complexity, which renders phenomenological modeling attempts quite laborious.

In parallel to the above, the design of new materials and products must take into account the important sustainability challenges of the modern industrial production paradigm, as well as the competitive environment and dynamic market demands that necessitate constant development and production-on-demand readiness. In this sense, the increasing interest in ML techniques for CPE applications comes as a natural consequence, since these techniques are specifically adapted to the increased complexity of these systems, as will be illustrated in the rest of this report.

There exist numerous excellent reviews of ML applications in various areas of chemistry and chemical engineering, as presented in Table 1. This review article, in addition to presenting an updated state-of-the art in the field, will focus on ML applications in the specific area of CPE over the last 20 years. Accordingly, particular attention will be paid to the design and discovery of new molecules and materials, the modeling of the relationship between process and product structure or properties, the prediction of chemical reactions and retrosynthesis and the support to sensorial analysis, via the prism of ML modeling approaches. In addition, a general guideline on the selection of the appropriate ML techniques, according to the characteristics of the problem under study, as well as a discussion about the advantages, limitations and challenges associated with these models are provided at the end of the article.

The rest of the article is divided into four main sections. Section 2, provides a background of ML categories illustrated with examples in CPE, the following Section 3, presents the state-of-the art of ML techniques in each of the aforementioned domains of CPE and, finally, Section 4 presents a critical discussion and the guideline for similar modeling attempts. The large number of abbreviations that are used throughout the discussion is listed at the end of the paper.

Table 1. ML reviews in different domains of chemistry and chemical engineering.

Domain	References
Molecular and material science	[2–13]
Drug design and discovery	[14–18]
Catalysis	[19–21]
Chemical synthesis	[22–24]
Chemical and process engineering	[25–27]
Additive manufacturing	[28,29]

2. ML: Background

2.1. Categories of ML Algorithms

ML algorithms are typically classified into four different learning categories, namely supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning [30,31]. These categories are defined by the configuration of the data set, on the basis of which the ML algorithm will attempt to identify mathematical correlations in the form of a model. The latter will then enable to solve the given problem. Different types of problems can be addressed within the different learning categories. These are briefly outlined in Table 2 and further detailed in the rest of this section.

- **Supervised learning**

This learning category is named "supervised" as a reference to a teacher who teaches a student the right answer for a given problem, taking into account the different factors (a.k.a. features) of the problem. When the student faces the same problem again with a new, but similar, set of features, he is then able to guess the right answer on the basis of the examples he learned from the teacher. However, if the new set of features is too different from the ones of the examples, the student's answer is more likely to be wrong.

In supervised learning, the data set is composed of N labeled examples (i.e., in the sense that the "correct" answer is provided along with the features), $\{(x_i, y_i)\}_{i=1...N}$ where x_i and y_i are, respectively, the input and the output vectors of the i^{th} example. The input vector contains the set of features, while the output vector is composed of the label(s), or the right answer(s), corresponding to this set of features. In the same way as the student learns from the teacher's examples, the supervised learning algorithm uses this data set to model the relationship between the features x and the labels y. The obtained model can then predict the label(s) for a new feature vector, provided that the latter is not too different from the ones of the examples as well as that the model has learned only the underlying trend of the data and not their noise. This is also referred to as the bias/variance trade-off [30].

There are two types of problems for which supervised learning algorithms are commonly employed: regression problems (the label is a continuous value) and classification problems (the label is a discrete value). Artificial neural network (ANN or NN), support vector machine/regression (SVM/SVR), Gaussian process (GP), decision tree (DT), random forest (RF), k-nearest neighbors (kNN), multivariate regression (MR) and logistic regression are examples of popular supervised learning algorithms.

Some of them are more suitable for treating regression problems (e.g., MR), others are more adapted to classification problems (e.g., logistic regression), while several of them can be used in both regression and classification problems (e.g., NN and SVM). The main principles of some popular supervised learning algorithms will be explained later in this article.

Table 2. Comparison of the different ML categories.

Learning Category	Training Data Set Configuration	Objective	Examples in Chemical Product Engineering	Examples of Algorithms
Supervised	Labeled data $\{(x_i, y_i)\}_{i=1...N}$	The algorithm describes the relationship between inputs x and outputs y	• Regression problem (continuous output): prediction of water-in-oil emulsion viscosity according to temperature, dispersed phase volumetric fraction, shear rate and oil properties (LSSVM) [32] • Classification problem (discrete output): classification of steel microstructure according to textural features and morphological parameters (SVM) [33]	ANN SVM/SVR GP DT RF kNN MR Logistic regression
Unsupervised	Unlabeled data $\{(x_i)\}_{i=1...N}$	The algorithm explores and extracts hidden patterns within the input features x	• Clustering problem: grouping of samples of tea according to their fermentation degree (HCA) [34] • Dimensionality reduction problem: dimension compression of process data to address high correlations between different variables and reduce the computational cost during the prediction of a polypropylene melt index using GP (PCA) [35]	PCA k-means clustering ANN HCA AE ICA GMM
Semi-supervised	Few labeled data with a large amount of unlabeled data	The algorithm explores the information hidden in unlabeled data in order to improve the prediction performance of the supervised learning model constructed with the labeled data	• Online prediction of Mooney viscosity in industrial rubber mixers [36] • Prediction of the thermal conductivity of polymeric composites filled with BN sheets [37]. • Others: [26,38–47]	ANN Generative models Graph-based methods Co-training Self-training Multiview learning
Reinforcement	Input data are the states and the feedback signals of environment; output is action	The algorithm learns an optimal policy that selects which is the best action to execute given the state of the environment	Control of polymerization processes [48,49]	Dynamic programming Monte Carlo methods Temporal difference

Different applications of supervised learning can be found in CPE. The authors in [32] used a least-squares support vector machine (LSSVM) to predict the water-in-oil emulsion viscosity according to four features: the temperature, dispersed phase volumetric fraction, shear rate and oil properties. The authors in [33] applied SVM to predict steel microstructure classes according to the textural and morphological features. The first application is a regression problem as the output is a continuous value (viscosity) while the second one is a classification problem since the output is discrete (microstructure class).

- **Unsupervised learning**

As its name implies, the learning here is "unsupervised", which means that the student is not taught by a teacher what is the right answer for different sets of features of a given problem. Instead, the student compares the features and attempts to determine if they present similarities. Accordingly, in unsupervised learning, the data set is composed of N unlabeled examples $\{(x_i)\}_{i=1...N}$, where x_i denotes again the input vector of the i^{th} example. The algorithm uses only these input vectors to build a model that explores and extracts hidden patterns within the features.

Among unsupervised learning problems, the most common ones are related to dimensionality reduction and clustering. On the one hand, dimensionality reduction is used for the compression of large data sets as a means to reduce the computational burden of the learning algorithm, as well as to eliminate eventual correlations between the features. Several unsupervised ML techniques also allow a representation/visualization of the data in a way that the sought patterns and correlations become more easily identifiable, not only by the algorithm but also by the user, thus, facilitating the analysis and comprehension of the problem.

Principal component analysis (PCA) is, by far, the most popular algorithm of this family, typically employed for the reduction of the dimensionality of the feature space in a precursor step of subsequent model development stages. On the other hand, clustering refers to the process of identification of existing clusters in the input data. The so-called clusters are groups of data that present a relative similarity with respect to a specific characteristic.

K-means clustering is one popular clustering algorithm, mainly due to its ease in application and its low level of mathematical complexity. Other unsupervised learning algorithms include autoencoders (AE), hierarchical clustering analysis (HCA), independent component analysis (ICA) and Gaussian mixture model (GMM), while ANNs find application in this category as well. The main principle of some of the most encountered algorithms will be explained later in this article.

Different applications of unsupervised learning can be found in CPE. Concerning the clustering problem, the authors in [34] used HCA to identify groups in tea samples according to their fermentation degree. As for the dimensionality reduction problem, the authors in [35] applied PCA to eliminate high correlations between different variables in the process data and, therefore, decrease the computational cost during the prediction of a polypropylene melt index using GP.

- **Semi-supervised learning**

In semi-supervised learning, the data set is generally composed of a small amount of labeled data and a majority of unlabeled data. The target is identical to supervised learning, but additionally the idea here is to explore the information hidden in large amounts of unlabeled data in order to improve the prediction performance of the supervised learning model constructed with labeled data. The premise here is that the enlargement of the data set, achieved by the addition of unlabeled examples, results in a more accurate representation of the probability distribution that the labeled data came from [31].

Semi-supervised learning has become popular in the process industry only recently, compared to supervised and unsupervised learning. Typical examples concern fault classification and quality prediction problems, in which the cost of labeling is high, thus, hindering the implementation of a fully labeled training process [26]. This increased cost is

mainly due to the fact that the acquirement of labeled data necessitates the implication of human experts effort and/or expensive analytical devices. On the contrary, unlabeled data is much cheaper and takes less effort to acquire from the process.

Several methods for semi-supervised learning have been applied in the literature, such as generative models, graph-based methods, self-training, co-training and multiview learning [26,38]. Some applications are listed here. Ensemble deep learning was used for quality prediction in industrial polymerization processes and in coal preparation processes [39,40]. The authors in [36] proposed a semi-supervised extreme learning machine (ELM) for online Mooney viscosity prediction in industrial rubber mixers. The authors in [41] applied bagging local semi-supervised models for the soft sensing of silicon content.

The authors in [42] performed probabilistic representation and the inverse design of metamaterials using a deep generative model with a semi-supervised strategy. The authors in [43] automatically detected faults for laser powder-bed fusion. The authors in [44] explored semi-supervised variational autoencoders for biomedical relation extraction. Semi-supervised learning was also applied in drug discovery for the prediction of drug function from chemical structure analysis [45]. The authors in [46] employed semi-supervised local kernel regression for the soft sensor modeling of the rubber-mixing process. The authors in [47] implemented a semi-supervised methodology for the operating condition recognition of multi-product pipelines.

A more detailed example of a semi-supervised learning application is that of [37] for the thermal conductivity prediction of polymeric composites filled with boron nitride (BN) sheets. Most thermal conduction models require many experimental results for calibration of the empirical parameters, which is time-consuming. In addition, there is still a lack of mature theory to build a systematic thermal conduction model with good accuracy and generalization performance. In this work, a co-training artificial neural network (Co-ANN) method was proposed to take advantage of the numerous unlabeled data to refine the thermal conductivity prediction.

Four inputs variables were considered, namely the thermal conductivity of polymer matrix, the diameter, aspect ratio and volume fraction of BN sheets. The labeled data set was first used to establish two ANN supervised regression models with different architectures. The average of the latter two was then used to pseudo-label the unlabeled samples. The following step was the confidence estimation of the pseudo-labels from the mathematical influence and thermal conductive behavior. This confidence estimation was compared to the lower limit of the labeling confidence, which was introduced to reduce noise interference and error accumulation brought by the use of the pseudo-labeled examples. This allowed selection of only the most confidently labeled examples in the augmented training data set for the ANN semi-supervised regression model.

Due to the augmented training data set and the introduced lower limit of labeling confidence, the obtained Co-ANN thermal conduction model remarkably improved the thermal conductivity prediction and showed the best accuracy and generalization performance compared to other theoretical models. This work represents a great potential in material design when no mature theoretical models are available and experiments are time-consuming.

- **Reinforcement learning**

In reinforcement learning, the goal is to train an agent to learn an optimal policy that selects which is the best action to execute, given the state of the environment or system. To do so, the agent interacts dynamically with its environment by executing actions, for different states of the environment, and readapting its behavior according to the positive (reward) or negative feedback (punishment) it will receive after each action. Therefore, the state of the environment and the reward or punishment signal can be considered as the inputs of this learning method and the action is the output. The optimal policy is obtained when the actions maximize the rewards.

Reinforcement learning has recently gained increasing popularity in control tasks in process industries, in robotics and gaming since AlphaGo, a computer program, managed

to defeat a professional Go player in 2015 [26,48–51]. As such, its principal application spectrum in process engineering is related to process control problems. The authors in [50] provided a review of the applications of reinforcement learning in industrial process control. Different types of algorithms exist, such as dynamic programming, Monte Carlo methods and temporal difference. At the same time, applications in CPE remain rare.

An example of reinforcement learning application is the work of [48] who proposed a polymerization reaction system controller based on deep reinforcement learning (DRL). The control is performed by simultaneously adjusting both the monomer and initiator flow rates to follow the target weight-average molar mass M_w optimal trajectory in a simulation environment. In this case, the agent is the DRL controller, the reactor system is the environment and the action is the combination of the control variables (i.e., the monomer and initiator flow rates).

The state that is an input for training the controller is composed of the current M_w and historical measurements. Indeed, current M_w is not a good representation of the current state of the environment, due to the huge time delay of the reaction and, therefore, is not adequate alone for predicting the future outcome. At each time step, the agent receives a reward from the environment. The reward contains the difference between the setpoint and measurement as well as a time term to adjust to the importance of reaching the setpoint at the end of the batch experiment. This term provides extra reward for reaching the set range when the reaction approaches the end of the reaction and increases the penalty otherwise.

The developed DRL controller was able to make a control policy for a process with multiple inputs, non-linearity, large time delay and noise tolerance. One advantage comparing to traditional controller is that the exploration is done in an automated manner. Additionally, no parameter tuning or real-time optimization is necessary, which makes the DRL controller easily adaptable to various systems and capable of controlling highly nonlinear systems and high frequency tasks.

2.2. Hybrid and Combinatorial Approaches

Common mathematical complexities encountered in CPE problems are non-linearity, large and multi-scale systems, long dynamics, uncertainties and high-dimensionality [52,53]. When there is, additionally, a lack of sufficient knowledge about the physical and chemical laws governing the system, it becomes very difficult and time-consuming to develop pure physico-chemical (i.e., knowledge-based) models to solve these problems. In these cases, hybrid models can represent an interesting solution. A modeling approach is typically characterized as "hybrid" when it combines techniques deriving from different families or categories. In this sense, the term is commonly employed to describe the combination of data-driven (or black-box) models with knowledge-based ones, in an attempt to exploit the forces of both model types.

In general, knowledge-based models describe the underlying phenomena of a process on the basis of prior knowledge and, as such, possess significant predictive capacity in a very large domain of application. On the downside, they demand a rather laborious development procedure and may result in an overall difficult-to-solve form (i.e., in terms of mathematical solution), especially when implemented to describe complex systems.

Data-driven models, on the other hand, are employed in an attempt to create a mapping between some selected input(s) and response(s) of the system, on the basis of available data. The form of the equations can be any mathematical expression, whose terms have no physical meaning. As such, they are typically very fast to develop and simple in structure, however, at the same time, suffer from a limited extrapolation capability (i.e., their application is restricted to the domain represented by the data) and a poor understanding of the mechanisms.

Accordingly, hybrid modeling approaches, combining both knowledge-based and data-driven characteristics display increasing popularity in solving problems where the mechanisms are too complex to be exhaustively described mathematically or where the relevant knowledge/understanding of the phenomena prevailing on a specific part, or range of con-

ditions, of the process is missing. Numerous relevant applications have been reported in the food industry [54–56], biopharmaceutical industry [57,58], cosmetic products design [59], catalysts design and discovery [21], reaction prediction [60] and polymer processes [61–64].

At the same time, it is also very common to combine different ML techniques, mostly in a sequential manner, within the framework of the same problem. Typical examples include the implementation of a dimensionality reduction technique prior to application of a regression or classification method, in order to reduce the feature space and select the most relevant inputs, thus, reducing the computational cost of the latter step [35,65–68]. Although the characteristics and the objectives of these combinatorial modeling approaches are quite different compared with those of the aforementioned hybrid models, they are sometimes used interchangeably in reported studies [52,65].

Several reviews of the applications of hybrid and combinatorial models are available in petroleum and energy systems engineering, multiscale material and process design and in separation processes [52,69,70]. The authors in [52] presented hybrid models as an alternative of data-driven models and first principle models in terms of knowledge of process, computational burden, data demand and extrapolation capabilities. In the same work, different types of hybrid model structures were also presented, such as serial and parallel configurations.

The authors in [69] highlighted the importance of hybrid methods in multiscale material and process design. Indeed, hybrid models are of great interest for tackling these complex and multiscale design problems where the material selection and process operation are strongly interacting and require consequently simultaneous material and process design. Concretely, material properties, which are computationally expensive to obtain, are generally described by data-driven models, while the well-known process-related principles are represented by mechanistic models.

The authors in [69] also presented a generic design methodology as well as the current limitations and future opportunities. Similarly, the authors in [71] combined the ML-based solubility model with first-principle absorption process models to perform integrated ionic liquid and process design for CO_2 capture.

3. ML in Chemical Product Engineering: State-of-the-Art

3.1. Current Challenges in Chemical Product Engineering and Role of AI/ML

Over the last three decades, the chemical industry has been continuously looking for opportunities to manufacture the necessary commodity chemicals as well as to convert them into higher value-added products [72]. In particular, the interest in these high value-added chemical products has become even more marked due to the competitive worldwide context and dynamic market demands. Chemical industries have to differentiate themselves by constantly developing innovative products as fast and as economically as possible while ensuring quality, performance and sustainable manufacturing [53,72–75].

This trend gave rise to CPE as a building block of chemical engineering. On the one hand, chemical engineering elaborates commodity chemicals, well known and best served by process design with a focus on the optimal and sustainable transformation of raw materials and energy into targeted products. On the other hand, the problems encountered in CPE aim at developing new and/or improved products based on customer needs and/or new technologies [53,76].

These products present a strong correlation particularly between the ingredients (composition and physico-chemical properties), the product (micro-structure, end-use macroscopic properties) and the processing conditions. A wide diversity of these products can be encountered in high-performance materials, semi-conductors, cosmetics, inks, pharmaceuticals, personal care products, household products and foods [77].

High value-added chemicals are characterized by their complexity due to their variety of structures, functions and compositions. Specialty chemicals (e.g., surfactants) are one category of high value-added chemicals and consist of pure compounds that, contrarily to commodity chemicals, are produced in small quantities and present a specific benefit

or function. Formulated products (e.g., cosmetic and food consumer goods), another important category, are combined systems with usually 4 to 50 components, which are often multi-functional and designed to meet the end-use requirements [73].

Therefore, the study of these chemicals is especially complex since the different ingredients interact with each other, and theoretical models cannot easily describe those interactions. The correlation between the composition of a mixture and its final properties also cannot be easily captured: even if the properties are based on physico-chemical principles, theoretical models are still far from predicting the performance or the properties of the mixtures as a function of the ingredients [78]. The development of these models also requires a sufficient theoretical understanding of the domain, which is costly and time-consuming.

This is where the application of AI and ML techniques comes into play as it can provide a means to modeling the complex relationships between ingredient characteristics, process conditions and product properties, without any *a priori* demand of substantial theoretical knowledge, based solely on data. However, the availability of data of sufficient quality and quantity is crucial and will be discussed further.

The functional molecules used as ingredients in formulated products also need to be designed, discovered and synthesized in order to reach the targeted properties. Even if the number of known molecules keeps increasing, the exploration of the chemical space still remains a great challenge. To give an idea of its vastness, the number of potential possible structures for small drug-like molecules is estimated to 10^{63}, while only 140 million molecules have historically been reported in chemistry [79].

Computer Aided Molecular Design (CAMD) methods have been widely used in the design of molecules (new or existing) that meet certain desired properties. However, the application ranges of these methods are most often limited to the available models, data and knowledge related to the currently known products and/or simple molecules [72]. Consequently, most chemicals are still designed by experiment-based trial-and-error approaches. In addition, experimentation to improve and create new products is limited due to time and cost limitations [78]. In this sense, data-driven methods, such as ML, could greatly help to discover new structure–property relationships.

For all the aforementioned challenges, AI techniques could greatly help the development of these complex products in reduced time and costs. This is the reason why AI has gained increasing popularity in CPE problems in the last two decades, in a similar manner as in chemical engineering problems [51].

3.2. Overview of ML Methods in Chemical Product Engineering

This subsection provides a general overview of the ML methods that have been implemented in CPE problems from 2000 to 2021. After an initial exposition of the overall picture, the thematic area of CPE is further distinguished into a number of application domains, such as molecular and materials science, polymer science, food industry, cosmetics and pharmaceutical industry and catalysis, and a number of relevant recent studies are reported for each domain. Given the huge amount of reported studies, as well as the expansion rate of the relevant literature (cf. Figure 1), it is virtually impossible to cover the subject exhaustively in a single review publication.

The reported data in this section are based principally on the analysis of approximately 150 selected articles and review articles, published in the aforementioned period, and their respective references (i.e., a total of approximately 1500 references). Overall, this literature review pointed out the prevailing use of supervised learning methods in CPE, occupying an overall percentage of the reviewed reports of 69% (Figure 2). Hybrid, unsupervised learning and combinatorial methods displayed also significant applicability, as they were found to be implemented in 11%, 10%, and 7% of the reviewed articles, respectively.

At the same time, the implementation of semi-supervised learning methods (2%) and reinforcement learning methods (<1%) is thus far marginal in these application domains. These findings are consistent with the reported observations of other reviews, which also emphasized the major use of supervised learning methods compared to unsupervised learning methods in

CPE or chemical engineering problems [2,21,26,70]. The interest in semi-supervised methods appears to be quite recent and displays an increasing trend, showing that this category of ML methods may become more significant for problems in the domain.

Figures 3–5 describe, respectively, the distribution of supervised, unsupervised and hybrid methods in CPE applications. The most popular supervised learning algorithms are ANN, SVM/SVR, RF/DT, GP and kNN at, respectively, 35%, 20%, 12%, 8% and 5%, while the most popular unsupervised learning methods are PCA, ANN, ICA, GMM and k-means clustering at, respectively, 36%, 12%, 10%, 9% and 7%. As for hybrid methods, knowledge-based techniques are mainly combined with ANN, SVM/SVR, MR, partial least squares (PLS) and GP, representing, respectively, 44%, 8%, 6%, 5% and 4% of the applications. Figure 6 displays in which CPE sector ML techniques are most employed within the reviewed literature.

When considering all ML categories (left figure), the prevailing sectors are materials science (26%), food industry (23%), process industry (17%) and molecular science (15%). As for supervised learning methods (right figure), they are predominantly applied in the same sectors with slightly different percentages. Figure 7 depicts, in more detail, the type of problems that are principally solved using supervised (top figure) or hybrid approaches (bottom figure), namely modeling; optimization; control and monitoring; design and discovery; support to sensorial analysis; and reaction prediction. As for unsupervised methods, they are mostly used for dimensionality reduction, data visualization and information extraction.

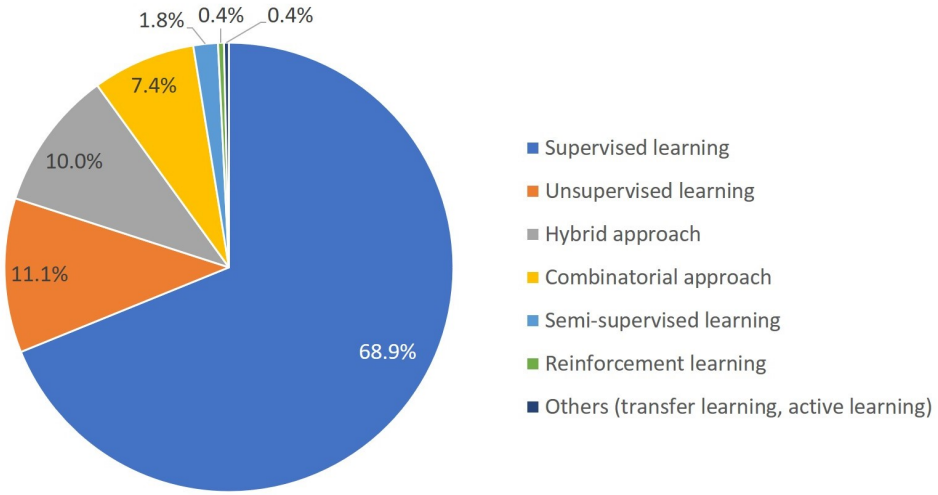


Figure 2. Distribution of the different ML categories in the reviewed CPE applications.

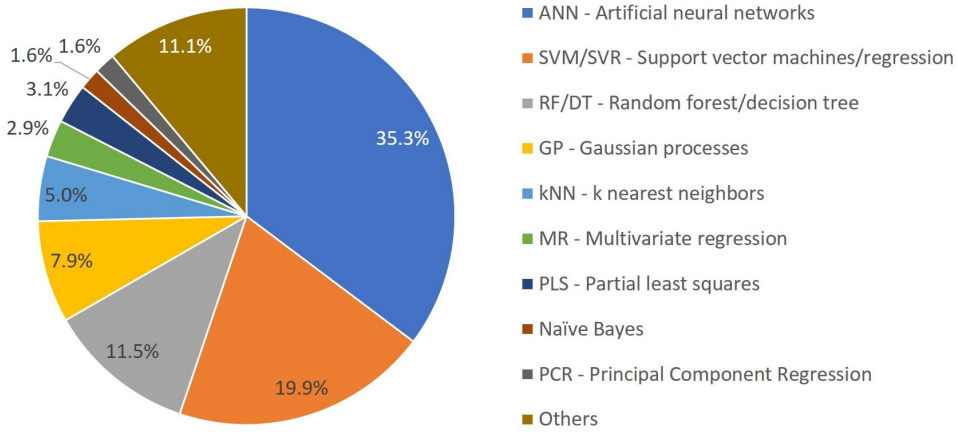


Figure 3. Distribution of supervised learning methods in the reviewed CPE applications.

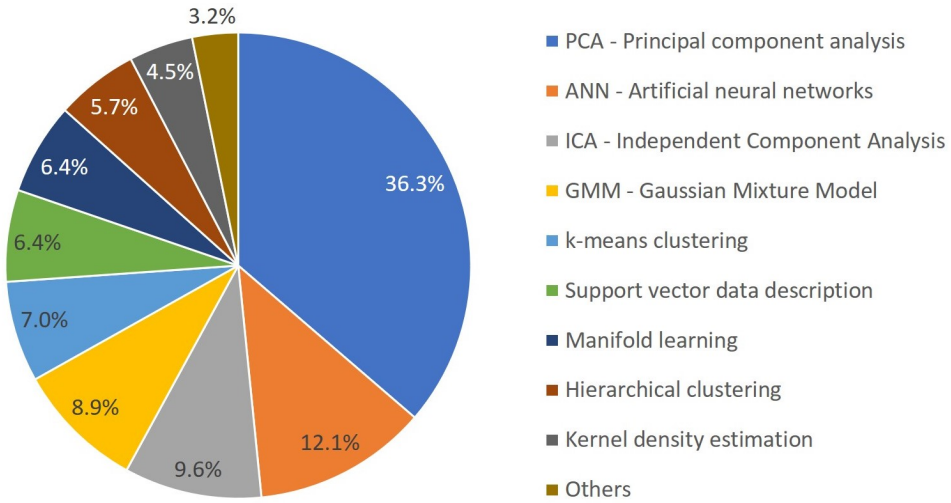


Figure 4. Distribution of unsupervised learning methods in the reviewed CPE applications.

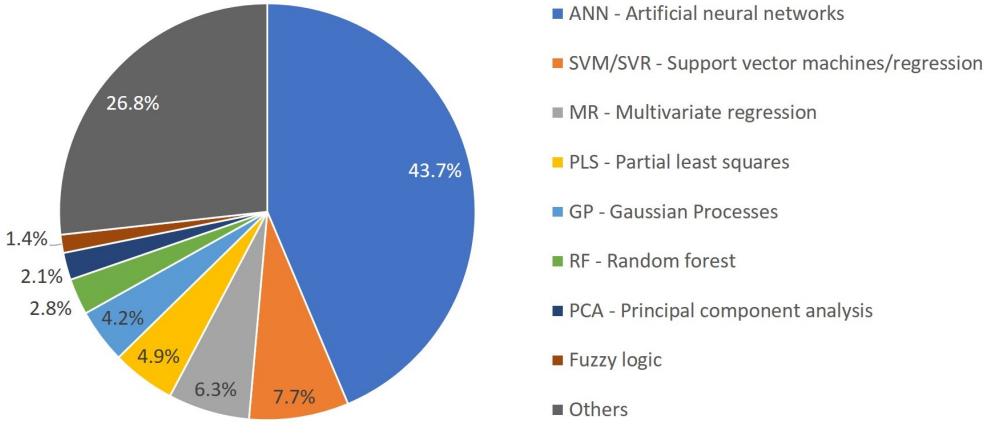


Figure 5. Distribution of ML methods, as part of hybrid modeling approaches, in the reviewed CPE applications.

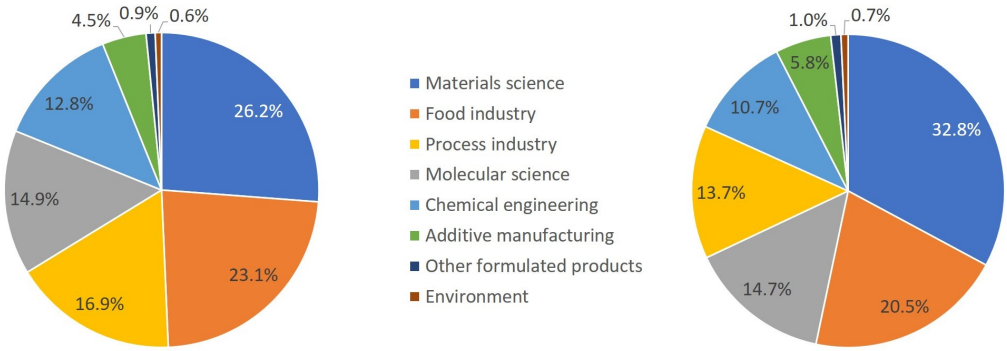


Figure 6. Distribution of CPE sectors in the reviewed ML applications. (**left**) All ML categories are considered. (**right**) Only supervised learning is considered.

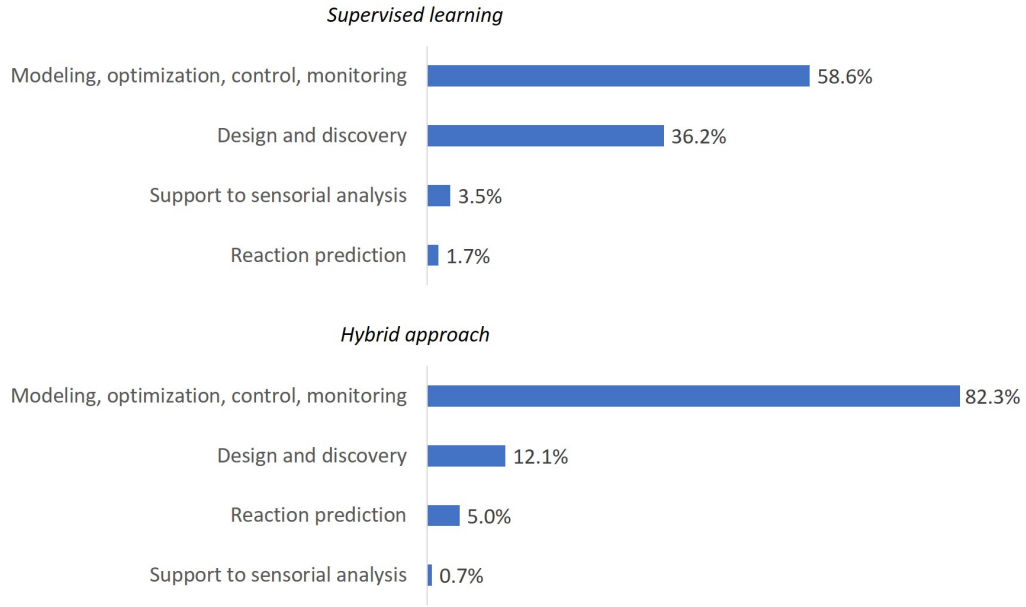


Figure 7. Distribution of CPE problem types in the reviewed ML applications. (**top**) For supervised learning methods. (**bottom**) For hybrid methods.

3.3. Popular ML Applications in Chemical Product Engineering Problems

- **Design and discovery of new molecules and materials**

One of the major applications of ML in CPE is the design and discovery of new molecules and materials referring, respectively, to the understanding and/or optimization of structure/property relationships, as well as to the exploration/screening of the large and high-dimensional chemical space with high throughput/autonomous techniques. Computer-aided techniques have shown their efficiency in these applications: for example, many organic chemicals-based products can be routinely designed through structure–property relationships [72].

However, as the chemicals are becoming increasingly complex, the existing models are not applicable anymore, and developing new models is costly and time-consuming as it requires establishing sufficient theoretical knowledge. For example, quantum mechanics (QM) methods (such as density functional theory (DFT) or semiempirical methods) or group contribution (GC) methods, which are commonly employed to calculate physicochemical properties, have shown limitations in their applicability to more complex and larger chemicals, which is often associated with their high computational costs [60,80,81].

As a result, new molecules and materials are often developed based on expert knowledge or trial-and-error experiments [75]. Nevertheless, ML methods could greatly help to extract quantitative structure–property or structure–activity relationships (QSPR or QSAR) from the collected data in cases where knowledge-based methods are limited [72].

The authors in [82,83] described the typical computational workflow for chemoinformatics QSPR-QSAR analysis using ML. This multi-stage processing is necessary as a chemical structure has to be converted into chemical information applicable for ML tasks. Thus, the first step is the encoding of the chemical structure, which consists of generating chemical descriptors (also called features) from the chemical structure. These descriptors are typically constructed in the form of chemical graphs, connection tables, linear text-based notations (e.g., SMILES, InChI and SMARTS) or fingerprints (i.e., vectors that indicate the absence or presence of a structural fragment/property) and contain the necessary information to provide as input to the model.

Additional details about these forms can be found in dedicated reported reviews, for ML applications [10,84–87]. This initial encoding stage is often carried out with the aid of

specific software packages, such as PaDEL and Python RDKit, which are open-source and publicly available.

The generated descriptors can vary from a simple molecular formula to complete 3-dimensional chemical conformation descriptors, including molecular weights, functional group counts, structural topology and geometry, hydrophobicity, solubility and electronic and steric properties, whose values may have been theoretically determined or experimentally measured. Deep learning (DL) methods (e.g., autoencoder/decoder, adversarial and convolutional neural networks (CNNs)) have greatly simplified the problem of generating mathematical descriptors to train ML models as they can transform simple representations of molecular entities (e.g., SMILES strings and linear text representations of organic molecules) to relevant descriptors internally [9]

The generated descriptors will usually contain "too much" information, with respect to the requirements of the modeling of a specific SPR/SAR. Hence, the second step in this ML workflow typically consists of a feature selection step by means of an unsupervised dimensionality reduction algorithm. This enables identification of the most significant features, from the high-dimensional features vector, to reducing the computational cost of the final step and to increasing the efficiency and the accuracy of the model. The last step is the learning phase where the mapping between the significant features (x) and the properties of interest (y) is established using a supervised learning algorithm. Examples of algorithms in such applications include Naïve Bayes, MR, kNN, RF, SVM and ANN.

Compared to physical models, such as quantum chemistry (QC), molecular dynamics (MD) simulations or early QSPR-QSAR methods, ML methods are more suitable for exploring non-linear SPR-SAR with high accuracy and precision, compared to DFT calculations, without necessitating any prior knowledge of their functional form. ML methods are also faster than DFT calculations by many orders of magnitude, thus, reducing the prediction horizon from several hours (or even days) to a few seconds [10].

This allows a significant acceleration of the discovery process, since the development of new materials via the conventional trial-and-error procedure requires several months, with the synthesis step being the major bottleneck [88,89]. Finally, ML methods are easily scalable to big data sets, such as libraries with large amount of candidates, without the requirement for extensive computational resources. A schematic summary of the materials discovery workflow in the age of AI was also given by [10].

There are two types of problems that can be formulated in the framework of QSPR/QSAR, namely the direct or forward design problems and the inverse design problems. The forward design relies on the prediction of targeted properties of molecules or materials, given their structure and descriptors. This is a typical straightforward modeling approach that follows the classical paradigm adopted also by other modeling techniques. The inverse design problem is formulated as the identification of the most-likely candidates that are prone to possess a targeted property value (or a set of properties/values).

This problem, which would be commonly formulated as a model-based optimization problem in the case of a phenomenological model, can be treated directly in the framework of a supervised ML approach by inverting the inputs and outputs of the model. In both cases, an experimental validation of the identified materials is necessary at the end of the procedure. The difficulty in modeling QSPR/QSAR depends partly on the molecule/material complexity. Indeed, complex materials are harder to study compared to small organic drugs or drug-like molecules.

For example, nanomaterials present distributions of shapes and sizes, the surfaces change dynamically depending on the environment in which they are embedded, and materials are often not well characterized [9]. Table 3 presents some examples of ML applications in direct and inverse design while Table 4 provides references of ML applications for the design and discovery of various molecules and materials.

Table 3. Applications of ML in the design and discovery of molecules/materials. Acronyms: AAE: adversarial autoencoder, ANN: artificial neural network, BL: Bayesian learning, BNN: Bayesian neural network, BO: Bayesian optimization, COSMO-RS: conductor like screening model for real solvents, DFT: density functional theory, DNN: deep neural network, GAN: generative adversarial network, GP: Gaussian process, GCPR: G-coupled protein receptor, LASSO: least absolute shrinkage and selection operator, LDA: linear discriminant analysis, MR: multivariate regression, NLP: natural language processing, PCA: principal component analysis, QM: quantum mechanics, RI: refractive index, RF: random forest, RL: reinforcement learning, RNN: recurrent neural network, SMILES: simplified molecular input line entry specification, SVM: support vector machine, and VAE: variational autoencoder.

References	ML Method	Inputs	Outputs	Data Set
• **Forward design (property/activity prediction from chemical structure)**				
[90] Fragrance	LDA, SVM	Molecules structural descriptors	Fragrance class (apple, pineapple, rose)	91 organic compounds with their fragrance class from database
[91] Cosmetics	ANN	Peptides	Anti-age properties	Data set from papers and patents (unstructured data) and from public databases (structured data), processed respectively by NLP and graph-based techniques
[92] Polymers	PCA/LASSO for data visualisation and feature reduction + GP for regression	Polymer relevant features	Refractive index (RI)	500 polymers from publicly available sources with their experimentally measured RI
[93] Polymers	GP + Lower confidence bound Bayesian optimizations	Molecular traceless quadrupole moment, molecule average hexadecapole moment	Glass transition temperature	60 polymers with their transition temperature from database
[94] Homogeneous catalysis	Hybrid: MR, Kernel ridge regression, RF, ANN and QM/DFT calculations	Energy, atomic, molecular, vibrational, structural descriptors (DFT)	Catalytic activity, reaction yield	4600–18,062 catalysts/reactions from libraries
[94] Heterogeneous catalysis	Hybrid: ANN, MR, RF, SVM, GP and QM/DFT calculations	Fingerprint features, structural and charge descriptors (DFT)	Adsorption, formation, binding, activation, reaction barrier energies, catalytic activity (DFT)	315–788 catalysts/reactions from libraries
[87] Molecules	Generative model for latent space creation (RNN, VAE, AAE, GAN, RL, BL and BNN) + predictive model for mapping latent variables and properties (RNN, RL, DNN, SVM, GP, BO and BNN)	Molecular representations (numerical, text-based or graph-based)	Physical, chemical or biological properties	5k–1800k molecules from databases
[67] Polymers	PCA/LASSO for data visualization and feature reduction + GP for regression	53–57 Relevant features from three hierarchical levels (atomic, block and chain)	Frequency-dependent dielectric constant, glass transition temperature	738 polymers and their 1210 experimentally measured properties at various frequencies
[95] Pharmaceutical compounds	GP and ant colony optimization algorithm (activity prediction followed by automated top scoring compounds picking from virtual combinatorial library)	Structure	Activity of ligand binding to 11 pharmaceutical relevant GCPRs (G-coupled protein receptor) drug target	3519 compounds with affinity annotations for 11 diverse GCPR targets (from libraries)

Table 3. *Cont.*

References	ML Method	Inputs	Outputs	Data Set
[83] Pharmaceutical compounds	ANN	Small molecules conformations	Energy of much larger systems	22 M small molecules conformations
[96] Polymers	GP (Polymer Genome)	Structure	Gas permeability	315 polymers and their associated 1501 permeability data
[97] Ionic liquids	ANN, SVM	Groups present in ionic liquid molecule	CO_2 solubility	10,116 CO_2 solubility data in various ionic liquids
[80] Cosmetics	Graph machine; Hybrid: ANN and COSMO-RS	SMILES, moments (COSMO-RS)	Viscosity	300 liquid compounds with known viscosities
[98] Polymers	GP, ANN, Kriging (Polymer Genome)	Polymer name, SMILES	Polymer properties	80–6721 polymers and associated properties obtained from first principles and experimental measurements
• **Inverse design (generation of candidates molecules/materials given target properties)**				
[88] Polymers	ANN	Lightweight, strong, chemical resistant	Candidates structures/patterns	Large database from experiments and simulations
[99] Ionic liquids	ANN	Ionic liquid maximized solubility	Top ionic liquids candidates for CO_2 capture	10,116 CO_2 solubility data in various ionic liquids
[100] Molecules	ANN SVM and Kernel ridge regression	Specifications, properties, reagents	Candidates (structures and products)	Not identified

Table 4. Other applications of ML for the design and discovery of diverse molecules/materials.

References
[101] Polymers
[102] Thin film nanocomposite membranes
[103] Heterogeneous, multicomponent materials
[104] Memristors materials
[105] Thermal functional materials
[106] Mechanical metamaterials
[107] Energy materials
[108] Photonic crystals
[109] Metal-organic nanocapsules
[110] Hydrogels
[111] Renewable energy materials
[112] Alloys
[113] Functional materials
[114] Polymers
[115] Ultraincompressible, superhard materials
[116] Materials for clean energy
[117] Photo energy conversion systems

- **Prediction of chemical reactions and retrosynthesis**

Reaction prediction (or forward reaction prediction) and retrosynthesis represent two of the main challenges of organic chemistry as they require chemists to have years of expertise. In a similar principle as in the direct and inverse materials design problems, the forward reaction prediction consists of predicting products given the reactants, reagents and reaction conditions. Inversely, retrosynthesis is the opposite procedure in which one seeks to predict the required reactants for the synthesis of a given product. The two problems are closely related, since a successful reaction prediction system can be used to validate a retro-synthetic proposal [118].

However, retrosynthesis is much more complex than forward prediction as several potential reactant combinations may lead to the synthesis of the same product (i.e., the equivalent of an optimization problem presenting multiple minima in the design space). Accordingly, this procedure is also often recursive until commercially available reactants are identified. The prediction of the reaction conditions (i.e., catalysts, reagents, solvents, temperature, pressure etc.) is also a challenge as the formulated multi-parametric design space presents the same characteristics as previously (i.e., multiple minima, discontinuities and high irregularity).

Overall, three different approaches have been used to computationally solve these two problems up to now, namely physical-based methods, rule-based experts systems and ML methods. Physical-based methods are based on simulations of the chemical reaction transition-state energies, primarily using QM. These methods result in very accurate predictions and provide in-depth understanding of the system but suffer from high computational costs and are limited to small molecules. Rule-based expert systems are computationally cheaper and have been very popular. They consist in establishing decision-making rules of human chemists using libraries of graphical rearrangement patterns or templates of chemical transformations.

However, they require a continuous time-consuming follow-up and update after any extension or modification of the database or the identification of a new rule or a conflict. Conversely, template-free methods are fully data-driven as no reaction templates are necessary. In this respect, ML methods can provide an interesting alternative, capable of responding to the aforementioned limitations, as they only require examples of reactions instead of encoded rules by experts and can significantly compress the simulation time. At the same time, their successful implementation depends highly on the availability, quantity and quality of relevant data.

Several sources of reaction information can be found in databases (not all publicly available), such as Reaxys, SciFinder, CAS, SPRESI, Beilstein and USPTO as well as the one from Lowe. For example, the latter is open-source and contains data for 1,808,938 reactions extracted from US patent grants and applications from 1976 to 2016 [119]. Another drawback in the application of ML methods in this domain is related to the fact that data sets include, as a majority, high-yielded reactions and only a limited number of negative examples of low-yielded or failed reactions, thus, severely biasing the available information that can be extracted from them. A comparison of the three approaches is given in Table 5.

Table 5. Comparison of the three major approaches in reaction prediction and retrosynthesis.

Method	Advantages	Limitations
Physical-based	• Accurate and generalizable • Deep understanding of chemistry	• Not adapted for high-throughput reaction prediction tasks and large systems (computationally expensive)
Rule-based expert systems	• Computationally cheap/quick prediction	• Requires a continuous time-consuming follow-up and update after any extension or modification of the database or the identification of a new rule or a conflict • Not generalizable (i.e., not encoded chemistry will not be predicted by a rule-based system)
Machine learning	• Only needs examples of reactions (i.e., no encoded expert knowledge required) • Low computational cost • Not limited to small molecules	• Lack of sufficient high-quality and publicly available data • Lack of negative examples (i.e., failed or low-yielded reactions)

Similar to the case of material design problems, the implementation of ML methods for the prediction (or retrosynthesis) of chemical reactions requires that the information extracted from the databases is transformed to a machine-readable format, before being injected to the model. To this end, molecular descriptors (structural, physico-chemical, electronic, topological) are once again employed in an adapted format, to describe the complete reaction procedure (in contrast to the description of a single molecule).

An example of such a reaction representation is via the use of reaction "fingerprints", defined by the difference of the respective descriptors of the reaction products and reactants. These so-called fingerprints are vectors of binary digits that describe the presence (1) or absence (0) of a certain group or substructure on the molecule. The most popular ML techniques used in reaction prediction and retrosynthesis are ANN and DL, as they are specifically suited for recognizing nonlinear relationships within large and diverse data sets.

Contrary to some early attempts based on expert systems that did not lead to practical applications, ML has been increasingly applied to support experts in reaction prediction and retrosynthesis over the last decade. In particular, there have been noteworthy contributions from the teams of Kayala [118,120], Coley [23,121], Segler [122–124] and Schwaller [119,125–127]. The authors in [24] provided an excellent review that was specifically focused on the state of the art in reaction prediction and retrosynthesis. It is expected that ML will be of great help in this area for reducing the time-consuming and costly experiments needed for validating a synthetic route. Various applications of ML in reaction prediction and retrosynthesis are given in Table 6.

Table 6. Applications of ML in reaction prediction and retrosynthesis. Acronyms: ANN: artificial neural network, DL: deep learning, GCN: graph convolutional network, HNN: hierarchical neural network, and SMILES: simplified molecular input line entry specification.

Application Category	References	ML Method	Inputs	Outputs
Reaction conditions prediction	[128]	HNN (classification and regression)	Reaction (difference between reactants and products fingerprints)	Reaction conditions (catalyst, solvent, reagent and temperature)
Ranking templates	[122,129,130]	ANN/DL/GCN (classification)	Reactants, reagents or product fingerprints	Most probable reaction template
Generating products	[119,131]	ANN/DL (encoder/decoder translation model)	Reactant SMILES	Product SMILES
Classifying reaction feasibility	[124]	ANN/DL	Product	Likely reactions
Predicting mechanistic pathway	[118,120]	ANN	Reactants, conditions, products	Reaction, mechanistic pathway
Ranking products	[121]	ANN	Possible reactions given reactants	Major product

- **Modeling and optimization of process–properties relationship**

Formulated and functional product properties are highly dependent on the prevailing process conditions during their synthesis and/or transformation steps. As a result, modeling the process–properties relationship is of paramount importance in CPE to ensure product quality and optimize production. The same general principles apply in this case as well; depending on the specific characteristics and complexity of the process, sufficiently accurate phenomenological models may exist or not, thus, making the necessity to resort to alternative data-driven models more or less pronounced.

Other factors playing a crucial role in this decision are the existing knowledge of the phenomena and the mechanisms governing the process, as well as the available resources in terms of time and/or budget limitations. Finally, data-driven techniques can be considered of specific interest, even in the case of existing knowledge-based models, when the simulation time is of importance (e.g., for online applications and optimization studies) or when the latter ones are highly dependent on ambiguous assumptions or on experimental characterizations that are difficult to acquire.

The state of the art highlights numerous applications of ML, especially in regression problems, for the prediction of product properties given the processing conditions. The inverse procedure, i.e., the prediction of processing conditions given the target properties is also encountered but less frequently, thus, providing the advantage to avoid complex inverse modeling and optimization procedures. Examples of applications of ML in process–properties modeling are summarized in Table 7 for various domains, such as polymer/material science, catalysis and the food/pharmaceutical/mineral/textile industries.

Depending on the application domain, diverse target properties are also predicted, including mechanical, structural and sensory properties, with respect to different process conditions, such as the temperature, time and composition. The sizes of the datasets used in these applications are relatively limited (i.e., typically inferior to 100 data) compared to other applications of ML. For example, in the aforementioned application domain of reaction prediction and retrosynthesis, the size of the databases ranges from several thousands to a few millions of reaction data. This is mainly due to the difficulties associated with the realization of experimental measurements in order to acquire the relevant process–properties data, which are time-consuming and costly.

These difficulties have also given rise to ML applications in soft sensor problems, which consist in predicting quality variables that are slow and/or difficult to measure directly, via the use of alternative process measurements that are faster and easier to acquire [54,132]. For example, in industrial polyethylene polymerization processes, it is

more interesting to relate the measurement of the melt index of the produced polymer, which is normally analyzed offline every several hours, with alternative process variables, such as temperature and pressure, which can be readily measured online with high frequency [39].

Another reason for this lack of abundance of representative data is related to the discontinuous nature of the processes that is often encountered in CPE, where the production specifications are frequently modified to manufacture products of different grades and with different properties. To overcome these limitations, larger data sets can be obtained directly from simulations (hybrid methods) [62,133], publicly available databases for the same system [56] or exploitation of relevant unlabeled data in combination with a semi-supervised ML method [39].

The polymer industry has a large number of ML modeling applications to display, within a process–properties relationship context, mainly due to the nature of the polymer molecules that are inherently complex (i.e., in terms of their macromolecular nature and diversity of structures and conformations). Indeed, the quality of a polymer product depends on a wide range of morphological and molecular properties, with direct implications for its end-use properties (i.e., physical, chemical, thermal, rheological and mechanical) and applications [134].

However, other important difficulties, specific to the polymerization processes, also explain the interest in ML techniques. Polymerization systems are commonly marked by a significant increase in the viscosity of the reaction medium, by several orders of magnitude, along the reaction, with direct implications on the control of the prevailing heat transfer rates as well as on the very mobility of the different macromolecules, thus, affecting the reaction rates. The mechanistic modeling of polymerization reaction kinetics can be extremely complex and time-consuming depending on the system. Indeed, these models contain a large number of differential equations, complex reactions occur simultaneously and many kinetic variables can be unknown or difficult to determine precisely.

In addition, the properties of the produced polymer products can be modified at-will by the addition of diverse materials, such as fillers and reinforcing agents, during different steps of the process. An example is the use of recycled tire powder to modify the mechanical properties of polystyrene [135]. Although the kinetic modeling of styrene radical polymerization is well-documented and relatively trivial, the presence of the recycled tire powder of variable composition in the system brings about a series of diverse effects on the evolution of the kinetic developments that are difficult to describe due to the current limited understanding of these mechanisms.

Table 7. Applications of ML in process–properties modeling. Acronyms: ANN: artificial neural network, C2V: code2vect, CFD: computational fluid dynamics, DBN: deep belief network, DoE: design of experiments, DT: decision tree, GP: Gaussian process, iDMD: inspired by dynamic model decomposition, LMNNR: large margin nearest neighbor for regression, MR: multivariate regression, PCA: principal component analysis, RF: random forest, sPGD: sparse proper generalized decomposition, and SVM/SVR: support vector machine/regression.

References	ML Method	Inputs	Outputs	Data Set
Polymer science				
[136]	ANN	Dwell time, oven temperature, tension applied on filaments	Yield, final properties of carbon fibers	Not identified
[39]	Semi- supervised: DBN and kernel learning	Reactor pressure/temperature, liquid level and catalysts flowrate	Melt index	1900 unlabeled + 310 labeled
[137]	ANN	Process parameters	Monomer conversion, average molecular weight and viscosity, reaction time, dispersion and thermal stability	Not identified
[138]	SVM	Temperature, feed rates, reaction time and catalyst quantities	Viscosity	120 labeled
[139]	ANN, SVR, GP	Injection speed/pressure, packing duration/pressure, mold temperature, cooling time, shot size, screw rotation speed, cylinder pressure, barrel temperature, coolant temperature and sensor measurements	Product quality (deformation, defects), melt state, process parameters, fiber orientation distribution, physical/mechanical properties, skin layer and surface roughness	Not identified
[35]	PCA + GP	Hydrogen concentration, feed rate and reaction temperature	Process conditions and product quality	300 labeled
[140]	ANN	Temperature and clay composition	Dynamic mechanical properties (storage modulus and loss tangent)	More than 1500 labeled
[141]	GP	Process parameters (position, constriction angle, channel width, polymer and solvent flows)	Product parameters (median length, median diameter and quality of fibers)	Not identified
[142]	ANN, C2V, sPGD, SVM, DT and iDMD	Material and process parameters (rotation speed, exit flowrate, temperature and compositions)	Properties and performance (Young modulus, yield stress, stress at break, strain at break and impact strength)	59 labeled
[61]	Hybrid: knowledge-based and C2V and sGPD	Flowrate and rotation speed	Torque, pressure, engine power and exit temperature	47 labeled

Table 7. *Cont.*

References	ML Method	Inputs	Outputs	Data Set
[133]	LMNNR, Nearest Neighbor Regression with adaptive metrics	Reation conditions (initiator concentration, temperature and time)	Monomer conversion and average molecular weight	337–414 labeled
[62]	Hybrid: knowledge-based and ANN	Reation conditions (initiator concentration, temperature and time)	Monomer conversion and average molecular weight	3363 labeled
[143,144]	ANN	Reation conditions (initiator concentration and temperature)	Monomer conversion, average molecular weight and mass reaction viscosity	Not identified
Food industry				
[54]	Hybrid: knowledge-based and SVR, SVM or ANN	Easy measurements (massecuite temperature/volume/level, vacuum degree, steam pressure/temperature and feeding rate)	Difficult measurements (mother liquor purity/supersaturation)	210 labeled
[56]	Hybrid: knowledge-based and RF	Food ingredients (selection and composition), processing conditions (baking time and temperature)	Sensory properties (color, crispiness and flavors)	446–462 labeled
[66]	Hierarchical clustering	Intrinsic characteristics of yogurt product	Brand and storage conditions	36 unlabeled
Pharmaceutical industry				
[145]	ANN/Fuzzy logic	Flowrates, frequency of vibration and concentrations	Microparticles properties (shape, oil content and distribution)	41 labeled
[146]	ANN/Fuzzy logic	Compositions, stirring speed	Properties of nanoparticles (size, size distribution, zeta potential, encapsulation efficiency and drug loading)	15 labeled
[147]	MR	Base equivalents, water equivalents and solvent loading	Dynamic profile of starting materials, product and key impurity	25 labeled
[148]	CFD and DoE and ANN	Dimensionless parameters based on material properties, concentration of the particles, viscosity of the injection solution and ratio needle diameter over the greatest dimension of the particles	Drug injectability	319 labeled

Table 7. *Cont.*

References	ML Method	Inputs	Outputs	Data Set
Paints				
[149]	ANN, MR	Formulation parameters	Thermodynamic and functional properties (elasticity, hardness and barrier properties)	Not identified
Catalysis				
[150]	ANN	Nominal silver concentration, pH, reaction time, actual amount of Ag attached on ZnO surface, initial contaminant concentration and light wavelength	Actual amount of Ag attached on ZnO surface and photodegradation performance	27–63 labeled
Minerals				
[132]	PCR	Fast and easy measurements (flowrate, pressure, temperature and spectra)	Slow and difficult measurements (composition, size distribution, mill load and equipment failure)	Not identified
Textile				
[151]	ANN	Process and structure parameters (bleaching or dyeing, bio-polishing, softening, emerizing, calendering, material and count of yarn)	Sensory properties (bipolar, surface and handle attributes)	23 labeled
Materials science				
[152]	SVR, ANN	Structure and process parameters (temperature, stretching ration and space velocity)	Mechanical property (Young's modulus and tensile strength)	30 labeled

- **Support for sensorial analysis**

Sensory evaluation has been widely applied in diverse industries, such as the food, cosmetic and textile industries for quality inspection, product design and marketing. Indeed, these industries have to propose diversified products that satisfy consumer preferences, and sensory attributes represent important factors to assess the quality and market acceptance of consumer goods. While appearance is rather easy to evaluate, the sensory qualities of a product are more difficult to evaluate, and its assessment is, therefore, a challenge to ensure high quality products. Up to now, the common practice to evaluate sensory attributes and to predict customer responses is through sensory panels, composed of humans (experts or not and trained or not), whose evaluation is considered representative of the general target population [153].

In this respect, the members of the panels use their senses to assess sensory properties, such as the color, aroma, taste of a product and skin feeling. However, sensory panel evaluation exhibits several drawbacks in terms of resources (costly, time-consuming, necessity to train some panels, a well-defined procedure to guarantee same sensory conditions for all panelists…) and also in terms of data characteristics and quality. In particular, data in sensory evaluation is subjective and uncertain and can contain inherent sources of misinterpretation (e.g., linguistic expressions) [154,155]. There is a general lack of reproducibility, standardization of the measurements and comparability between evaluations of different panels [156]. All these reasons make sensory panel assessments ill-adapted to routine quality evaluation.

The state of the art highlights diverse applications of ML methods to assist the complex sensory evaluation of products, with a specific emphasis on food products, and to study the impact of process, microstructure or chemical compositions on sensory attributes. Accordingly, ML methods have been increasingly applied in problems where more classical methods, deriving from the field of chemometrics, were typically implemented. Chemometrics uses multivariate analysis and statistical methods, such as analysis of variance (ANOVA), PCA, partial least squares (PLS), MR or principal component regression (PCR), to analyze multivariate data, instrumental or not.

Chemometrics methods, such as ANOVA or PCA, are most often applied before ML methods to select and compress the original data [156–164]. Feature extraction methods such as Fourier analysis or Si-PLS are also used for extracting relevant information or optimal spectral interval from high dimensional spectra measurements [156,158].

This step is particularly important when working with spectral data as it prevents many irrelevant or redundant spectrum variables from being introduced and, therefore, decreases the complexity and size of the variable space and improves the precision of the model [164]. Supervised ML methods, such as ANN, SVM and RF, are then applied to link the sensory properties with the process parameters, the ingredients or the microstructure of the tested products.

Several works compare the use of AI techniques and classical approaches. The authors in [155] compared the uses of classical computing techniques, mostly based on statistics and multivariate analysis (PCA, generalized procrustes analysis and generalized canonical analysis), with AI in sensory evaluation. They found that AI methods had better ability in solving specific and human-related problems using both linguistic and numerical data processing.

They can also take into account nonlinear relationships as well as the specificity of sensory data and uncertain evaluation conditions. In another study, the authors in [159] used SVM and concluded that regression methods, such as PLS, were not able to capture consumer preferences, due to the so-called "batch-effect" which was not compatible with the consideration of the consumer ratings as absolute assessments. In other words, the ratings should be treated as relative data, in relation to the rest of the evaluated objects in the same "batch" of the sensory analysis.

Great effort in sensory evaluation has been made in an attempt to replace the subjective sensory evaluation with the use of more objective, robust and reproducible instrumental

and analytical measurements, in view of overcoming the associated uncertainty, imprecision and time demand of the classical sensory evaluation procedures [155,164]. Analytical methods, such as near infrared (NIR), Fourier transform infrared (FT-IR) and Raman spectroscopy analyses, or other instrumental methods combined with ML or chemometrics algorithms, have been used as efficient methods to quickly evaluate the biochemical or physical characteristics of food products and decode their correlations to sensory attributes.

The authors in [157] successfully applied ANN to predict chocolate physico-chemical properties and sensory descriptors based on specific absorbance values of NIR spectra. This rapid (one spectral measurement takes only 15 to 90 seconds) and non-destructive method represents an alternative to consumer panels in determining the sensory properties of chocolate in a more accurate, cheaper manner using chemical parameters. In another application, the authors in [158] reported a sensory analysis of puffed snacks crispiness-related freshness level, for various humidity levels, via the recording of mechanical and acoustical data.

In this study, ANN and SVM were selected for their ability to provide models that are more similar to the way sensory integration takes place in humans, in comparison with algorithms relying on linear relationships. Contrary to that, [158,165] opted for the implementation of RF for predicting wine olfactory characteristics from the volatile organic compound content as measured by gas chromatography–mass spectrometry (GC-MS), in order to gain interpretability in the final model.

ANN has also been employed, in combination with AdaBoost, to improve the prediction performance of the sensory attributes of rice wine from the NIR spectra [156]. The authors in [153] combined PLS regression and SVM to predict pork meat sensory attributes (tenderness, juiciness and chewiness) and quality grade group, on the basis of Raman spectra, while [166] reported the development of an ANN-based predictive model of several sensory descriptors of beer, using NIR spectra.

The prediction of the smell impression from the physicochemical properties of a molecule represents an important breakthrough for the cosmetic, beverage and food industries. A large number of experienced panelists are usually needed to create the desired odors through trial-and-error approaches in these industries. At the same time, the mass spectra of physicochemical properties can be used to represent structural information of the constituting molecules of a product that can be correlated with its odor.

The dimensionality reduction step, in these applications, is often carried out via the use of non-linear ML methods, such as an autoencoders neural network (AENN), which prevents the loss of information compared to the aforementioned classical chemometrics techniques. The authors in [167] utilized an AENN in the dimensionality reduction process of mass spectra of molecules from the NIST (National Institute of Standards and Technology) database and performed the clustering of descriptors by natural language processing to predict the odor category of chemicals. The authors in [66] used AENN combined with SVM for the prediction of yogurt preferences using sensory attributes.

The authors in [164] compared several linear and nonlinear dimensionality reduction (kernel PCA, sparse PCA, local tangent space alignment, PCA and multidimensional scaling) and regression methods (relevance vector machine, back-propagation ANN and PLS) for estimating the sensory quality of tea from NIR spectra. In this application, nonlinear methods displayed better performance due to existing non linearity in tea components and NIR spectra. In addition, their increased performance was also attributed to the structure of human sensory organs that act as an extension of the highly-complex central nervous system, displaying high degrees of sensitivity and specificity.

Another application of ML methods in the field of sensorial analysis is to evaluate the impact of processing conditions on sensory drivers of liking. The authors in [162] used different ML techniques, namely RF, gradient boosted tree and extreme learning machine, to predict the sensory drivers of cheese manufactured with milk subjected to conventional thermal processing and to ohmic heating, an emerging thermal technology in the dairy industry. Hybrid approaches are also encountered in this field as well. The authors

in [56] combined RF with mechanistic models to predict food sensory characteristics (color, crispiness and flavors) with respect to the ingredients (selection and composition) and the processing conditions (baking time and temperature).

In addition, in the food industry, where ML techniques find widespread application, the cosmetics and textile industries are also interested in similar ML and/or hybrid approaches to support sensorial analyses. For example, in the field of cosmetics, the authors in [168] employed an ANN-based surrogate model, as part of an integrated optimization-based cosmetic formulation methodology, including the implementation of mechanistic models and heuristics, to predict the sensorial rating of cosmetic products given their recipes and microstructures. The authors in [151] also used ANN and fuzzy logic for tactile sensory property prediction from the process and structure parameters of knitted fabrics.

The interest in ML for supporting sensorial analysis is expected to rise given its capacity in treating the associated complex interactions that impact product quality and sensorial attributes. A typical example concerns wine, where important quality traits, such as the sensory profile and color are a product of complex interactions between the soil, grapevine, environment, management and winemaking practices. ANN has been shown to be an efficient tool in assessing these complex interactions and predicting wine sensory properties from NIR spectra and from weather and water management information [163]. This example is illustrative of the way AI can present an opportunity for winemakers to adjust vinification techniques in order to obtain a more consistent wine style, predict market and consumer acceptance for pricing adjustments and provide better description of wines on labels for accurate information to consumers.

Finally, it is worth noting new emerging technologies that, combined with ML, enable performing analyses in a more standardized and rapid manner, such as robotic pourers with computer vision, electronic tongue or nose sensors or low-cost NIR spectroscopic devices and color sensors, attachable to smartphones with applications in food and beverages [163,169,170].

4. Guidelines for Applying ML in Chemical Product Engineering Problems

While the previously presented state of the art outlined the variety of ML methods applied in diverse applications of CPE for solving different types of problems, this section aims at providing some general guidelines for applying ML in relevant problems. In this respect, the principle of the most commonly encountered ML methods is briefly presented, along with their main advantages and limitations. Then, the discussion is extended to several aspects related to the interest of employing data-driven methods, the challenges that are frequently encountered in the process and some rules of thumb that may serve as indicators in the selection of the ML technique in relation to the problem characteristics and data configuration.

4.1. General Principle of Some Popular ML Methods in Chemical Product Engineering

According to literature review, most popular supervised ML methods in CPE seem to be ANN, SVM and GP. As for unsupervised methods, PCA is the most widely used.

- **ANN**

ANNs are widely encountered both in chemical engineering and in CPE problems. This is a family of methods that are based on the principle of the capacity of the human brain neurons to "learn" and repeat a specific action, given relevant stimuli as input. ANNs can be effectively considered as systems of interconnected calculation nodes, i.e., the so-called "neurons", that exchange messages amongst them. A typical ANN generally consists of an input layer (containing a number of nodes equal to the number of input variables), an output layer (containing one or more nodes to represent the output(s)) and one or more intermediate layers in between (also called "hidden" layers).

Each of these intermediate layers is also composed of a number of neurons, which are connected to the neurons of the adjacent layers. Each connection is associated to a weight. A neuron is a processing unit, which transforms the input data (i.e., the sum of all inputs

arriving at the neuron multiplied by their corresponding weights plus a bias term) to the output by an activation function. Examples of activation functions are given by [52].

The values of the network parameters (i.e., weights) are adjusted during the learning step on the basis of a set of training data through an iterative process of minimization of the distance between the predictions of the ANN and the responses (i.e., labels) of the data set. Common learning algorithms are Levenberg–Marquardt, gradient descent, quasi-Newton method (BFGS) and scaled conjugate gradient. The most important parameters to consider when designing an ANN are the number of hidden layers, the number of neurons in each hidden layer, the activation functions and the training algorithm.

Note that the number of neurons in the input and output layers are explicitly defined by the problem characteristics. The optimal network architecture, in terms of its number of layers and neurons, is usually defined based on a trial-and-error approach, by evaluating the performance of ANNs of different architectures. This evaluation is often based on the value of the mean squared error (MSE) between the network outputs and data set labels.

The role of the activation function of the hidden layer(s) is to introduce nonlinearity to the overall model, thus, increasing its capacity to simulate highly complex, non-linear response surfaces [152]. Accordingly, after the training step, the network can be used to perform diverse tasks, such as regression, classification and dimensionality reduction. ANNs may also include internal "recycle" connections (i.e., recurrent networks), providing them the ability to adapt better to dynamic problems and to time-series data. In addition, multiple "stacked" ANNs may also be combined, in different manners, to exploit the uncertainty in their predictions.

Finally, "deep" ANNs (i.e., "deep learning") can also be constructed with the combination of several layers of sequential convolution and pooling operations, thus, allowing a more efficient feature learning process of highly-dimensional problems; however, their analysis exceeds the framework of this report. A more detailed description of the different types of ANNs (such as single or multi-layer perceptron (MLP), recurrent neural networks (RNNs) and convolutional neural networks (CNNs)) can be found in [25,30,52,171].

Although the exact form of the mathematical model that is produced by a neural network is quite complicated, as it contains a large number of terms (i.e., relevant to its number of neurons and connections), developing a NN model is greatly simplified by the use of a number of dedicated libraries and softwares (e.g., Matlab toolbox, Python-C++ Scikit-Learn/TensorFlow/Keras, R and Weka) that offer a more user-friendly way of handling them [172].

The power of ANNs resides exactly in their ability to approximate any linear/nonlinear function by learning from observed data, presenting, at the same time, inner structure flexibility, adaptability, and a dynamic nature. As such, they are commonly employed in problems where the form of the response surface is entirely unknown (i.e., a lack of previous knowledge) and/or displays a highly nonlinear, multi-dimensional (i.e., in terms of the features), complex nature.

ANNs bypass the necessity to explicitly define the nature of the terms of the derived mathematical model, as is commonly the case for classical experimental design data-driven approaches. At the same time, in order for a ANN to be sufficiently accurate, significant amounts of data are often required. In this sense, they are recommended mainly for applications in which large volumes of data are either available or easy to generate [10,173]. In addition, ANNs are also prone to overfitting, thus, requiring specific attention during the learning process (more details about the phenomena of over/under fitting of ML methods are given in paragraph 4.4), which also presents some inconvenience due to the existence of multiple local minima.

- **SVM**

SVM is another popular ML technique that is most commonly employed for classification analysis. The model finds the hyperplane that separates the input data into distinct classes in a way that the "margin" (i.e., defined by the decision boundary lines and containing the hyperplane in the middle) between the classes is maximized. To define this

margin, SVM uses only those points, among all input data points, that are located closest to an eventual decision boundary. These are the so-called "support vectors", explicitly dictating the optimal position of the separating hyperplane by maximizing the distances between them and the hyperplane. When a linear hyperplane (i.e., a line or a plane) is adequate to separate the classes, the model is linear.

In the opposite case, where the data cannot be considered as linearly separable, SVM can still be applied, in combination with a projection of the data set to a space of different dimensions (typically a higher-dimensional projection is employed), through the use of "kernels". This mapping procedure transforms the data set into a linearly-separable one, thus, making the use of SVM again possible. There exist different types of kernel functions, such as Gaussian, polynomial, radial-basis functions and sigmoidal.

A major advantage of SVM, with respect to ANN, is its robustness in reaching a global optimum during the training (or learning) process. In fact, the problem of maximizing the margin is formulated as a quadratic constrained optimization problem, presenting a global minimum [26,174]. It can also perform with high precision and generalization with a small number of training samples, high dimensional and noisy data [52,152]. In fact, SVM uses a penalty hyperparameter to treat misclassification cases of noisy data, containing errors of labeling or outliers [31]. Among the drawbacks of the method is that its prediction performance is highly dependent on the suitable setting of its parameters, such as the kernel function, regularization parameter and insensitive loss function [174].

In addition to classification problems, SVM is also employed in regression problems, in which case it can also be referred to as support vector regression, SVR. The principle of the method, when applied to regression problems, is the same as in classification, i.e., a hyperplane is sought in this case as well. However, this time, the points that are considered lie within the decision boundary, and the goal is to have as many points as possible located on (or around) the hyperplane, which serves as the regression function. More details about SVM and SVR can be found in [25,30,31,175].

- **GP**

GP is a supervised ML method that has also been increasingly used in CPE for regression problems. GP is based on the description of a probability distribution over functions, defined by a mean function and a covariance matrix. Concretely, for a given set of training points, a typical regression procedure would require assuming the form of the function that best describes them before identifying the values of the parameters of this function. As ANN overcomes the difficulty that is posed when this functional form is unknown by employing a highly-complex mathematical expression to substitute it, GPs propose to select the best-fitting function out of a large number of different candidates.

In this sense, GP defines a prior over functions (i.e., a large number of candidate functions for the given problem) that is gradually transformed to a posterior over functions, once enough data have been presented to algorithm. GPs are considered to be "non-parametric" techniques, in the sense that their scope is to identify a specific set of parameters that are a priori posed by the form of a known function and, rather, to identify the function itself.

In order to define the probability distribution over functions (i.e., prior or posterior), GPs are based on the premise that these functions are jointly Gaussian, characterized by their mean and their covariance matrix. The mean represents the most probable output of the data, while the covariance serves as a measure of the smoothness of the functions [176]. Accordingly, two inputs that are considered similar (or close to each other) will also, most probably, produce similar outputs. GPs will, therefore, improve the confidence of their predictions as they receive new data, allowing them to better identify the posterior over functions, and new inputs that are similar to the existing ones.

As such, one of the great advantages of this method is its rigorous treatment of uncertainty, which helps in avoiding overfitting problems [177]. Its inherent preference to smooth functions is an additional factor that increases its generalization capacity, as fitting artifacts are avoided [176]. The probabilistic structure of GP can successfully incorporate

the noise information and provide uncertainty prediction result (confidence interval) for the process, which is very helpful for quantifying prediction reliability in problems of evolving conditions or wide operating ranges [35].

Consequently, GPs are employed in numerous applications, in addition to typical regression problems, such as surrogate model identification, dynamic experimental design and manifold learning-based modeling [177–179]. On the other hand, a major drawback in the application of GPs is their high computational demands when dealing with large data sets [10,177,180]. In fact, as the implementation of the method requires the continuous manipulation of the covariance matrix, whose size is directly proportional to the size of the data set, the computational cost increases. As such, GPs are rather more adapted to problems involving small data sets, which is in contrast to ANNs that require abundant data. More detailed descriptions of GP can be found in [181].

- **PCA**

PCA is, by far, the most commonly used algorithm for dimensionality reduction problems. The aim is to transform the original set of variables to a new set of uncorrelated variables, called principal components, without significantly reducing the relevant statistical information contained in the data. As such, it aims at finding an optimal trade-off between information loss and simplification of the problem. The method is based on the principle of projecting data from a k-dimensional space to a n-dimensional one, thus, reducing the considered coordinates.

For example, when a set of data is projected from 2D to 1D, their surface is reduced to a single line, just as a projection from 3D to 2D reduces the "cloud" of points to a plane. Accordingly, PCA will identify k principal components, orthogonal to each other (i.e., uncorrelated), on which to project the original data, in a way that the projection error will be minimized. This error is defined as the sum of squares of the segments between each point and their projected counterparts. Once all principal components have been identified, a reduced number of them will be commonly retained for the rest of the analysis, while the rest will be ignored. This number will depend on the amount of statistical information they contain, described via the percentage of the total variance that they are able to explain.

This step of the selection of the main principal components is, therefore, crucial to the successful application of the method, since a low number of selected principal components may bring about a significant loss of information. Inversely, retaining too many principal components reduces the efficiency and the whole intent of applying PCA in the first place. Commonly considered thresholds for this selection vary between 90% and 99% of the total variance.

PCA is usually employed as a preprocessing stage on the data before a regression or classification problem. It is highly recommended for multi-variable/high-dimensional problems to reduce the computational time and memory demands and to avoid overfitting issues that may be caused by the consideration of an excess number of features, often correlated among them [26,65]. PCA is also very useful in visualizing high dimensional data sets (i.e., by plotting them with respect to the first two or three principal components), which turns out to be extremely helpful in better understanding the data set before applying an appropriate regression or classification technique.

The main limitation of PCA is that it performs linear transformation of the variables, thus, somewhat limiting its capabilities with respect to nonlinear dimensionality reduction methods, such as ICA. PCA is very well documented in several statistics textbooks and dedicated reports [25,30,31].

- **Other ML methods**

For more details on the previously described ML methods as well as other popular ML methods, a plethora of dedicated textbooks and articles is available in the open literature. In addition, over the last years, numerous online courses from highly recognized researchers and institutions have become freely available, serving as excellent entry points to the world of ML algorithms. An indicative list of such sources is given in Table 8.

Table 8. Sources for an introduction to the fundamentals of ML algorithms.

MOOCs	Books, Articles
• Machine Learning—Coursera • Deep Learning Specialization—Coursera • Machine Learning with Python—Coursera • Advance Machine Learning Specialization—Coursera • Machine Learning—EdX	[30] [31] [182] [25] [175]

4.2. Interest of Data-Driven Methods

Given the popularity that ML methods have gained over the last years, their implementation to problems has become a standard. Indeed, increasing researchers are tempted to apply ML techniques, driven either by their popularity or their undeniable capacities. However, ML methods are not always interesting to apply in all problems, nor are all types of ML techniques adapted to all kinds of problems [30,182].

A very good reason to resort to data-driven techniques in general or to ML techniques in particular may be related to a substantial lack of knowledge and understanding of the mechanisms underlying a system or process, in combination with the lack of resources (i.e., in terms of time, budget, personnel, infrastructure etc.) to seek this knowledge via phenomenological approaches. For example, in sensory evaluation, the problem of linking process parameters or ingredients to the sensorial properties involves the investigation of the unknown interactions of a large number of factors with each other, along with the understanding and description of their effect on the neurological responses of the human brain, a task that is still far from being trivial with the current scientific knowledge.

Another reason for opting for ML techniques may be related to the fact that the gain in understanding of the mechanisms is not difficult to reach but simply less interesting in comparison to other aspects of the study. Such aspects may concern the need for automation, speed or simplicity of the developed model. Accordingly, in the previously reviewed applications of chemical reaction predictions and retrosynthesis, the approach that is based on the coding of reactions rules is too laborious and limits the flexibility, automatisation and extensibility of these models to a point that it becomes more interesting to resort to data-driven techniques. ML can accelerate the prediction of the properties of new molecules, without the necessity to systematically make use of computationally expensive approaches, such as MD and QM methods.

Finally, ML techniques are specifically adapted in dealing with the complexity that is associated with multi-parametric, multi-dimensional, non-linear problems. The very fact that their operating principle is founded on the treatment of data makes them particularly suitable in identifying correlations and patterns that are distinguishable, or even comprehensible, by the human brain via mechanistic approaches. In this sense, their application to problems associated with the exploration of new domains and the seeking of unexploited information on the frontiers of scientific knowledge is specifically interesting. A typical example is the implementation of ML techniques for the analysis of new, unexplored areas of the chemical space for the discovery of new materials, molecules or reactions.

At the same time, ML methods possess their own share of limitations and drawbacks, both as a class of methods and as individual techniques, that must not be overlooked when considering their implementation to a specific problem. One of the most important issues is related to the availability, quantity and quality of data; however, this is addressed separately in a following section.

However, in a related topic, one must consider carefully the required resources, both for the data collection, cleaning and treatment, as well as the resources required for the training of the ML algorithm, which can be quite substantial in certain cases (e.g., in ANN and DL applications) before engaging in an ML application. In no case should ML methods (or AI methods as a whole) be considered "magic-tools" that will provide all the answers with the simple push of a button.

In addition to the above, ML methods, as data-driven techniques, inherit all the relevant drawbacks, such as poor understanding, as well as limited extrapolation capacities. As such, although they can be used to gain insight to the way different features affect an output or interact among them, they are not strictly capable of providing a deep understanding of the phenomena, the mechanisms and the driving forces behind the observations. Furthermore, their application domain is normally limited by the range in which the data set that was used for the training of the algorithm can be considered representative.

Any extrapolation outside this domain by no means guarantees equal prediction accuracy by the trained model. Accordingly, even though ML methods are often advantageous when applied to problems characterized by fluctuating conditions (e.g., in production monitoring for online quality control), due to their ability to quickly identify and adapt to the transitions in the input, this presupposes that these fluctuations have been, at a certain moment, part of the training data set that the model has "seen" before (i.e., in the case of supervised learning methods).

It is important to consider these aspects before implementing a ML technique to a given problem, in order to have a clear sight of the objectives of the modeling study, the capacity of the selected method (or combination of methods) to completely or partially contribute in reaching these objectives, as well as of the associated limitations and drawbacks of the developed model. This will eventually allow a maximum exploitation of the tremendous capabilities of these very powerful techniques.

4.3. Challenges and Solutions

Several challenges frequently encountered in ML applications in CPE are discussed in the following paragraphs, namely the availability and quality of data, the difficulty of chemical data representation and the lack of understanding. The initiatives implemented to address these challenges are also presented. Similar discussions about different application fields, such as synthesis planning, materials and drug discovery, can be found in other reported studies as well [11,23,183–185].

- **Data**

Any data-driven technique can only be as good as the data it uses. However, the choice of a representative data set of "good" quality and "sufficient" quantity is crucial to the performance and the reliability of the developed model. In computer vision and natural language processing areas, which have benefited from the AI-related research over the last decades, data are often abundant, publicly available and simple to acquire [79,186].

On the contrary, in CPE, data is more expensive to generate and rarely publicly shared due to confidentiality and competitiveness reasons. In addition, the uncertainty related to some types of generated data can be extremely variable, thus, creating additional drawbacks in their common utilization in shared databases. These elements are only some examples of the numerous challenges related to the data, as a constitutive unit of any ML application.

According to the above, the first big challenge concerns the data availability and extraction capacity. The scientific literature contains huge amounts of experimental and theoretical data in disorganized and unstructured forms, such as text, figures and tables. Since the task of efficiently extracting them in machine-readable form cannot be performed manually, text mining algorithms have been developed and are often used.

However, the lack of a generally-applicable standard along with the ambiguity and variability in the conventions that are met between different scientific domains (and even sometimes within the same scientific area), limit the universality of the developed dictionary and rule-based approaches [10]. Nuances of language and unconstrained diversity of figures and tables inhibit automated interpretation and extraction by text mining algorithms [186].

When data are obtained from experiments, their number is often limited. On the contrary, simulation-generated data or data registered in available databases (e.g., UPSTO for chemical reactions) typically reach much larger quantities. The availability of data is

also more or less dependent on the application area. For example, publicly available data are less abundant in the organic materials and polymer research domains, compared to inorganic materials and drug design [10,68,79,187]. Examples of databases can be found for organic materials [10], inorganic materials [11], chemicals [87], materials [188] and molecules and solid materials [5].

This lack of data is quite frequent in CPE problems. To overcome this limitation, the scientific community is looking for ML methods that are specifically adapted to limited-data problems, such as kernel methods, low-variance models with feature reduction capabilities, multi-process modeling and transfer learning [189–192]. An example is given in [10], where a DNN, implemented in organic materials design, updates its initial weights from a large data set, derived from a similar domain to the target problem, and then fine tunes its weights using a smaller, dedicated data set, thus, learning the subtle characteristics that are specific to the targeted application.

Another approach to deal with this absence of large data sets is the implementation of semi-supervised learning techniques. Accordingly, unlabeled data can be pseudo-labeled by the ML model, established on the basis of the available limited amount of labeled data, thus, forming an augmented data set. Finally, active learning is another interesting technique that is frequently employed when the acquisition of labeled examples is expensive [11,173,193]. In this case, the model learning process initiates using relatively few labeled examples. At a second stage, the algorithm examines the obtained preliminary results and selects a sub-sample of the unlabeled data on the basis of their potential contribution to the learning process.

These are subsequently annotated, often by the intervention of human-experts or via classical experimental techniques, and the obtained samples are added to the labeled training set to rebuild the model. This cyclic procedure continues until some convergence criterion or limiting condition has been reached, such as a satisfactory model accuracy or a maximum number of annotations due to budget or time limitations [31,173].

Finally, it should be noted that the current trend in research, intensively promoted over the last years by numerous funding organizations and research institutions, encouraging data sharing within the scientific community, is expected to greatly improve the aforementioned limitations [187]. Other solutions for improving data-sharing practices, such as the use of publication standards, Google's data set search or specialized consortium creations, are also evoked [186].

In addition to their availability, another significant issue is the quality of the data. A typical example is found in the area of chemical reaction prediction and retrosynthesis, in which applications of ML are relatively recent. The available data are often incomplete, especially with regards to the reactions conditions (i.e, solvents, temperature and catalysts), which are not always specified, despite their significance to the reaction output (i.e, products and yield of the reaction). In addition, databases contain principally high-yielded reactions, while failed or low-yielded reactions are often not reported as they are considered "failed" attempts.

However, these "negative" data contain a wealth of information that is as important as the "positive" data, since they can serve in the identification of undesired domains of the feature space (i.e., in contrast to leaving this space largely non-characterized). This aspect is also encountered in the design, discovery and synthesis domain [10]. Another sector where data quality is not guaranteed is polymer science. As explained by [187], many polymer-related databases are being established and improved. However, some imperfections of current databases still limit the widespread applications of polymer informatics.

For example, a lack of databases containing processing details or significant experimental information, that may be unintentionally or intentionally omitted, is frequently observed. As such, additionally to encouraging the sharing of data, chemical communities also need to insist on the importance of sharing negative data as well as any significant piece of information, related to experimental conditions. Related initiatives about the

automation and standardization of experimental data collection procedures can be found in reported studies [186].

Another data-related challenge concerns the difficulty in chemical data representation, in combination with the complexity governing the selection of the molecular features that can be directly associated to the sought molecular properties [79]. For example, two very similar molecules presenting stereoisomerism can have significantly different properties.

In this case, a simple two-dimensional representation of the molecules will ignore this important difference between the molecules, creating two training examples of identical features but different outputs, with detrimental implications in the training of a supervised ML model. However, three-dimensional representations require important computational resources and can be subject to uncertainty generated from conformation prediction, ligand orientation and structural alignment [82,194].

Given the importance in any data-driven modeling technique to incorporate the maximum amount of features that are relevant to the desired output, the correct identification of these relations between the molecular features and properties becomes of paramount importance. In addition to their correct identification, this domain presents another difficulty in the representation of the identified features, derived from the large variety of possible molecular notations (e.g., SMILES, SMARTS, InChI and fingerprints) and structural/functional characteristics (e.g., atom coordinates, bond distances, bond rotation and vibration frequencies).

In this sense, the SMILES notation has been widely used due to its compactness and intuitive aspect. However, it cannot be implemented to describe certain chemical families, such as organometallic compounds and ionic salts [84]. SMARTS is an extension of SMILES for substructure search and can specify different isotopes or bond types. InChI generates unique/canonical SMILES but its back-tracing to the original molecular graph is not always guaranteed. Chemical data representation, therefore, remains an important challenge where intensive research is being conducted. For example, DL methods are becoming increasingly popular as tools to obtain molecular representations and build more powerful models [86].

- **Lack of understanding**

Despite the significant growth in the application of ML techniques, as shown earlier in Section 1, a part of the scientific community and, more importantly, the economic sector, is still reluctant in their adoption. One of the principal reasons behind this skepticism is the lack of interpretability, explainability and transparency of ML methods [195]. In the case of ANN and DL for example, the complex architecture of the networks and the form of the resulting mathematical expression make it extremely difficult to identify which inputs impact the outputs the most or the least and in what way.

As such, although these methods make it possible to scale the modeling of extremely large and complex data sets very rapidly, they do not allow a clear traceability of the reasons that lead the developed models to behave the way they do in their predictions. As such, they create a source of hesitation in their acceptance, as their performance is not clearly founded on established principles, nor can it always be rationalized on the basis of obvious correlations.

In this respect, in parallel to applying ML methods, understanding how the algorithms work and "decoding" their decisions is a field that is increasingly gaining attention within the scientific community, in an attempt to ensure the consistency of their outcomes and increase the confidence on their implementation. The authors in [196] presents some tools, specifically dedicated to the task of decoding and rationalizing ANNs. One such tool consists in wiring more transparent models directly into the connections of a ANN in order to increase the external control over its procedures.

Another approach is based on the perturbation of the inputs of a network and a parallel monitoring and analysis of the subsequent deviation in its responses, to identify and understand its activation flow. To overcome the lack of transparency of black-box models as well as the lack of interpretability (coming from highly parameterized models

with arbitrary choice of hidden states), the authors in [190] proposed the implementation of very simple ML models with handcrafted features and the evaluation of the cost-benefit relationship associated with the model shrinkage.

For example, there have been recent efforts to develop visualization tools to help to monitor the gain produced by the addition of extra layers to DL models. At the same time, a greater contribution from expertise and knowledge-driven approaches (e.g., in hybrid models), as well as the implementation of posterior consistency checks, can also greatly contribute in the increase of the interpretability and the control of the way these techniques work. To maintain a certain understanding of a system, it is also generally preferred, whenever possible, to model what is known with phenomenological models and the unknown part with ML methods or, in other words, to prioritize hybrid methods.

4.4. General Guidelines for the Selection of a ML Method

Unfortunately, a clear recipe or guide for the selection of the appropriate ML method for a given application does not exist. However, on the basis of the aforementioned characteristics and limitations of the different techniques, it is possible to distill a number of general good-practice rules that may serve as initial guidelines throughout this selection process. These rules are by no means explicit or novel and should be considered in combination with the specific characteristics of the problem at hand.

The first thing to consider is the necessity and interest in the implementation of a ML method with respect to alternative knowledge-based or different data-driven approaches according to the discussion presented in Section 4.2. Once the objectives of the study have been clarified and the implementation of a ML technique has been identified as an interesting approach, there are several other factors that need to be considered before homing in on a specific technique. Note that, whenever phenomenological models are available, hybrid modeling approaches should be pursued.

As ML methods are data-driven methods, the characteristics of data, such as their type (i.e., labeled or unlabeled), their amount and their structure (i.e., text, table, molecules etc.) will greatly influence this choice. As such, if the data is labeled or not will determine whether the selection should be directed toward a supervised or unsupervised learning method, respectively. In the intermediate case of the existence of both labeled and unlabeled data, semi-supervised methods should be preferred.

Furthermore, if the labels of the data are continuous values, a regression technique will be in order, while discrete labels will require the implementation of a classification technique. The structure of the data will also influence the choice toward certain types of methods. While all methods are generally compatible with numerical data (such as vectors and tables), ANN and especially DL methods will be more adapted to more complex data structures, such as texts and images.

The amount of data can also be used to facilitate the selection of the proper ML technique. As a general rule of thumb, it is considered that higher amounts of data are related to better ML model performance. This is especially true for ANN and DL, which require large data sets due to their increased number of parameters. A frequently asked question concerns the number of data necessary to consider a data set large enough for solving a problem.

Although there is no straightforward answer to this question, it should be taken into account that the necessity of large data sets is directly related to the complexity of the formulated model, which, in turn, is proportional to the complexity of the problem (including the complexity of the data). At the same time, it is important to remember that the above general rule of requirement of large data sets is not applicable to all ML techniques. In fact, as discussed previously, certain ML methods, such as SVM, GP, kNN and kmeans clustering, are rather more adapted to small data sets, which makes them excellent candidates for problems of limited data availability.

Several other aspects can also be considered for the selection of the appropriate learning algorithm, such as the sophistication level of the associated mathematical princi-

ples of the method, the training speed, the prediction speed and the non-linearity of the problem [31]. For example, concerning the first of these aspects, some ML methods are easier to use and to explain to a non-expert audience, such as kNN, linear regression and decision trees, in comparison to more sophisticated methods, like ANN, DL and SVM.

This can significantly increase their attractiveness toward occasional users or when the explanation of the implementation of the ML technique is part of the scope of the study (e.g., for educational purposes). The training speed can also be a decisive factor, in combination with the available computational resources. For example, the training of large ANNs of different structures, as part of the network architecture optimization, may require several hours, or even days, to accomplish.

In parallel to the selection of the ML method, another important decision that displays a direct effect on the performance of the developed model concerns the selection of the features. Ideally, all possible factors that may have an influence on the selected response should be included in the features list of the problem. However, since this information is rarely known *a priori*, there is a tendency to include as many features as possible in the training of the model in order to increase the possibility of capturing underlying relations and effects. This strategy may result in actually decreasing the capacity of the model to generalize its predictions, due to the so-called "overfitting" phenomenon, where the over accumulated noise in the data is "learned" by the model [194,197].

This problem is particularly intense with ANNs [30]. At the same time, the computational demands of the model increase along as well. To overcome this problem, the necessary amount of uncorrelated features can be selected on the basis of existing knowledge, whenever available, or by using dimensionality reduction methods, such as PCA or AE, prior to the learning step. Regularization or data partitioning into different data sets to be used for the model training, validation (i.e., to check the convergence of the training process or to tune some hyper-parameters of the model) and testing (i.e., to check the performance of the model on a fresh data set, once the training process has been concluded) are also efficient counter measures against overfitting.

Inversely, overlooking or omitting an important feature in the training process, in view of keeping the model simple and reducing the probability of overfitting, will impact the model performance as well since the model will be too simple to learn the underlying structure of data and/or incapable of identifying the complete relations between input and output, thereby, resulting in the opposite situation of underfitting. The above general guidelines are presented in the form of a decision tree in Figure 8.

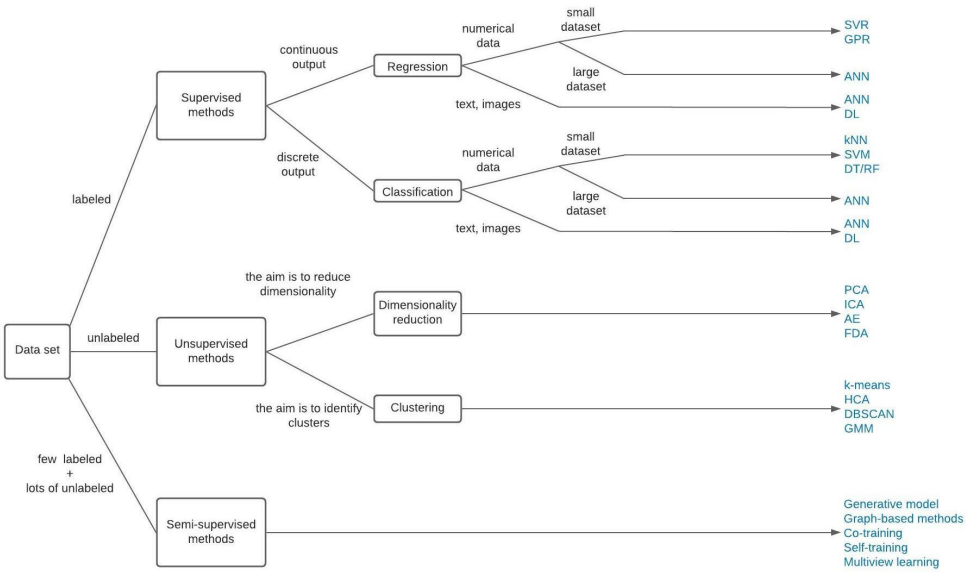


Figure 8. General guidelines for choosing appropriate ML methods.

5. Conclusions

Over the last decade, AI and ML techniques have been increasingly applied in CPE in order to solve the numerous complex challenges: the complexity of the structure-process–properties–ingredients interplay of the products and the necessity to quickly discover and constantly develop new molecules, materials, reactions and properties. In the present work, special emphasis was given to four selected domains, namely the design/discovery of new molecules/materials, the prediction of chemical reactions/retrosynthesis, the modeling of processes and the support for sensorial analysis.

Applications in the first two domains are relatively recent and intensive compared with the two others. The development of DL during the last decade enabled the tackling of extremely complex problems characterizing these first two domains, such as the exploration of the vast chemical space for both small organic compounds in the pharmaceutical industry and in materials.

More generally, the state of the art highlights the wide diversity in terms of the data characteristics among the different domains but also among the applications of a given domain. This provided a plethora of alternative ML approaches for the various problem types and data characteristics. Supervised, unsupervised and hybrid methods were found to be the most frequently implemented in CPE.

In addition, even if each domain displayed specific challenges, several common challenges could be identified, such as the ones related to data (i.e., data availability, data quality and chemical data representation), which are predominant in CPE as they are relatively more expensive and time-consuming to generate, with respect to other research domains. They are also rarely publicly available due to strong confidentiality and competitiveness limitations. This has lead to a significant growth in the use of ML methods that are specifically adapted to small data sets as well as to the development of massive data standardization and sharing initiatives.

Finally, even though a precise guide indicating the optimal ML method to use for a given problem does not exist, some guidelines are still provided here based on the problem constraints as well as on the characteristics of the available data.

Author Contributions: Conceptualization, C.T. and D.M.; literature research and analyses, C.T.; writing—original draft preparation, C.T.; writing—review and editing, all. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by MESRI (Ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation), France.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AAE	Adversarial AutoEncoders
AE	AutoEncoders
AENN	AutoEncoders Neural Network
AI	Artificial Intelligence
ANN	Artificial Neural Network
ANOVA	ANalysis Of VAriance
BFGS	Broyden–Fletcher–Goldfarb–Shanno
BL	Bayesian Learning
BN	Boron Nitride
BNN	Bayesian Neural Network

BO	Bayesian Optimization
BPNN	Back-Propagation Neural Network
C2V	Code2Vect
CAMD	Computer Aided Molecular Design
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
Co-ANN	Co-training Artificial Neural Network
COSMO-RS	COnductor like Screening MOdel for real solvents
CPE	Chemical Product Engineering
DBN	Deep Belief Network
DFT	Density Functional Theory
DNN	Deep Neural Network
DL	Deep Learning
DoE	Design of Experiments
DRL	Deep Reinforcement Learning
DT	Decision Tree
ELM	Extreme Learning Machine
FFNN	Feed-Forward Neural Network
FT-IR	Fourier Transform InfraRed
GAN	Generative Adversarial Network
GC	Group Contribution
GC-MS	Gas Chromatography–Mass Spectrometry
GCN	Graph Convolutional Network
GMM	Gaussian Mixture Model
GP	Gaussian Process
GCPR	G-Coupled Protein Receptor
HCA	Hierarchical Clustering Analysis
HNN	Hierarchical Neural Network
ICA	Independent Clustering Analysis
iDMD	inspired by Dynamic Model Decomposition
InChI	International Chemical Identifier
kNN	k-Nearest Neighbors
LASSO	Least Absolute Shrinkage and Selection Operator
LDA	Linear Discriminant Analysis
LMNNR	Large Margin Nearest Neighbor for Regression
LSSVM	Least Squares Support Vector Machine
MD	Molecular Dynamics
MDPI	Multidisciplinary Digital Publishing Institute
ML	Machine Learning
MLP	Multi-Layer Perceptron
MR	Multivariate Regression
MSE	Mean Square Error
NIR	Near InfraRed
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NN	Neural Network
PCA	Principal Component Analysis
PCR	Principal Component Regression
PLS	Partial Least Squares
QC	Quantum Chemistry
QM	Quantum Mechanics
QSAR	Quantitative Structure–Activity Relationship
QSPR	Quantitative Structure–Property Relationship
RBF	Radial Basis Function
RF	Random Forest

RI	Refractive Index
RL	Reinforcement learning
RNN	Recurrent Neural Network
sPGD	sparse Proper Generalized Decomposition
SMARTS	SMILES ARbitrary Target Specification
SMILES	Simplified Molecular Input Line Entry Specification
SVM	Support Vector Machine
SVR	Support Vector Regression
VAE	Variational AutoEncoders

References

1. Mitchell, T. *Machine Learning*; McGraw-Hill: New York, NY, USA, 1997.
2. Butler, K.T.; Davies, D.W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine learning for molecular and materials science. *Nature* **2018**, *559*, 547–555, doi:10.1038/s41586-018-0337-2.
3. Elton, D.C.; Boukouvalas, Z.; Fuge, M.D.; Chung, P.W. Deep learning for molecular design-A review of the state of the art. *Mol. Syst. Des. Eng.* **2019**, *4*, 828–849, doi:10.1039/c9me00039a.
4. Pilania, G. Machine learning in materials science: From explainable predictions to autonomous design. *Comput. Mater. Sci.* **2021**, *193*, 110360, doi:10.1016/j.commatsci.2021.110360.
5. Zhou, T.; Song, Z.; Sundmacher, K. Big Data Creates New Opportunities for Materials Research: A Review on Methods and Applications of Machine Learning for Materials Design. *Engineering* **2019**, *5*, 1017–1026, doi:10.1016/j.eng.2019.02.011.
6. Schmidt, J.; Marques, M.R.; Botti, S.; Marques, M.A. Recent advances and applications of machine learning in solid-state materials science. *NPJ Comput. Mater.* **2019**, *5*, 1–36, doi:10.1038/s41524-019-0221-0.
7. Westermayr, J.; Gastegger, M.; Schütt, K.T.; Maurer, R.J. Deep integration of machine learning into computational chemistry and materials science. *arXiv* **2021**, arXiv:2102.08435v1
8. Himanen, L.; Geurts, A.; Foster, A.S.; Rinke, P. Data-Driven Materials Science: Status , Challenges , and Perspectives. *Adv. Sci.* **2019**, *6*, 1900808, doi:10.1002/advs.201900808.
9. Winkler, D.A. Chapter 9 Machine Learning at the (Nano)materials-biology Interface. In *Machine Learning in Chemistry: The Impact of Artificial Intelligence*; The Royal Society of Chemistry: London, UK, 2020; pp. 206–226. doi:10.1039/9781839160233-00206.
10. Bennett, S.; Tarzia, A.; Zwijnenburg, M.A.; Jelfs, K.E. Chapter 12 Artificial Intelligence Applied to the Prediction of Organic Materials. In *Machine Learning in Chemistry: The Impact of Artificial Intelligence*; The Royal Society of Chemistry: London, UK, 2020; pp. 280–310. doi:10.1039/9781839160233-00280.
11. Zhuo, Y.; Tehrani, A.M.; Brgoch, J. Chapter 13 A New Era of Inorganic Materials Discovery Powered by Data Science. In *Machine Learning in Chemistry: The Impact of Artificial Intelligence*; The Royal Society of Chemistry: London, UK, 2020; pp. 311–339. doi:10.1039/9781839160233-00311.
12. Ramprasad, R.; Batra, R.; Mannodi-Kanakkithodi, A.; Kim, C.; Pilania, G. Machine learning in materials informatics: Recent applications and prospects. *NPJ Comput. Mater.* **2017**, *3*, 54, doi:10.1038/s41524-017-0056-5.
13. Chen, A.; Zhang, X.; Zhou, Z. Machine learning: Accelerating materials development for energy storage and conversion. *InfoMat* **2020**, *2*, 553–576, doi:10.1002/inf2.12094.
14. Zhu, H. Big data and artificial intelligence modeling for drug discovery. *Annu. Rev. Pharmacol. Toxicol.* **2020**, *60*, 573–589, doi:10.1146/annurev-pharmtox-010919-023324.
15. Yang, X.; Wang, Y.; Byrne, R.; Schneider, G.; Yang, S. Concepts of Artificial Intelligence for Computer-Assisted Drug Discovery. *Chem. Rev.* **2019**, *119*, 10520–10594, doi:10.1021/acs.chemrev.8b00728.
16. Lo, Y.C.; Ren, G.; Honda, H.; Davis, K.L. *Artificial Intelligence-Based Drug Design and Discovery*; Intech: London, UK, 2019; doi:10.5772/intechopen.89012.
17. Brown, N.; Ertl, P.; Lewis, R.; Luksch, T.; Reker, D.; Schneider, N. *Artificial Intelligence in Chemistry and Drug Design*; Springer Nature Swirzerland AG: Cham, Swirzerland, 2020; doi:10.1007/s10822-020-00317-x.
18. Klambauer, G.; Hochreiter, S.; Rarey, M. Machine Learning in Drug Discovery. *J. Chem. Inf. Model.* **2019**, *59*, 945–946, doi:10.1021/acs.jcim.9b00136.
19. Schlexer Lamoureux, P.; Winther, K.T.; Garrido Torres, J.A.; Streibel, V.; Zhao, M.; Bajdich, M.; Abild-Pedersen, F.; Bligaard, T. Machine Learning for Computational Heterogeneous Catalysis. *ChemCatChem* **2019**, *11*, 3581–3601, doi:10.1002/cctc.201900595.
20. Ma, S.; Kang, P.L.; Shang, C.; Liu, Z.P. Chapter 19 Machine Learning for Heterogeneous Catalysis: Global Neural Network Potential from Construction to Applications. In *Machine Learning in Chemistry: The Impact of Artificial Intelligence*; The Royal Society of Chemistry: London, UK, 2020; pp. 488–511. doi:10.1039/9781839160233-00488.
21. Yang, W.; Fidelis, T.T.; Sun, W.H. Machine Learning in Catalysis, From Proposal to Practicing. *ACS Omega* **2020**, *5*, 83–88, doi:10.1021/acsomega.9b03673.
22. Nair, V.H.; Schwaller, P.; Laino, T. Data-driven Chemical Reaction Prediction and Retrosynthesis. *Chimia* **2019**, *73*, 997–1000, doi:10.2533/chimia.2019.997.

23. Coley, C.W.; Green, W.H.; Jensen, K.F. Machine Learning in Computer-Aided Synthesis Planning. *Acc. Chem. Res.* **2018**, *51*, 1281–1289, doi:10.1021/acs.accounts.8b00087.
24. Haywood, A.L.; Redshaw, J.; Gaertner, T.; Taylor, A.; Mason, A.M.; Hirst, J.D. Chapter 7 Machine Learning for Chemical Synthesis. In *Machine Learning in Chemistry: The Impact of Artificial Intelligence*; The Royal Society of Chemistry: London, UK, 2020; pp. 169–194. doi:10.1039/9781839160233-00169.
25. Commenge, J.M. Big Data et Intelligence Artificielle pour le Génie des Procédés 2021. Available online: https://hal.univ-lorraine.fr/hal-03107557/document (accessed on 10 August 2021).
26. Ge, Z.; Song, Z.; Ding, S.X.; Huang, B. Data Mining and Analytics in the Process Industry: The Role of Machine Learning. *IEEE Access* **2017**, *5*, 20590–20616, doi:10.1109/ACCESS.2017.2756872.
27. Yan, Y.; Borhani, T.N.; Clough, P.T. Chapter 14 Machine Learning Applications in Chemical Engineering. In *Machine Learning in Chemistry: The Impact of Artificial Intelligence*; The Royal Society of Chemistry: London, UK, 2020; pp. 340–371. doi:10.1039/9781839160233-00340.
28. Wang, C.; Tan, X.P.; Tor, S.B.; Lim, C.S. Machine learning in additive manufacturing: State-of-the-art and perspectives. *Addit. Manuf.* **2020**, *36*, 101538, doi:10.1016/j.addma.2020.101538.
29. DebRoy, T.; Mukherjee, T.; Wei, H.L.; Elmer, J.W.; Milewski, J.O. Metallurgy, mechanistic models and machine learning in metal printing. *Nat. Rev. Mater.* **2021**, *6*, 48–68, doi:10.1038/s41578-020-00236-1.
30. Géron, A. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2019.
31. Burkov, A. *The Hundred-Page Machine Learning Book*; Andriy Burkov: Quebec City, QC, Canada, 2019.
32. Nasery, S.; Hoseinpour, S.; Phung, L.T.K.; Bahadori, A. Prediction of the viscosity of water-in-oil emulsions. *Pet. Sci. Technol.* **2016**, *34*, 1972–1977, doi:10.1080/10916466.2016.1233248.
33. Gola, J.; Webel, J.; Britz, D.; Guitar, A.; Staudt, T.; Winter, M.; Mücklich, F. Objective microstructure classification by support vector machine (SVM) using a combination of morphological parameters and textural features for low carbon steels. *Comput. Mater. Sci.* **2019**, *160*, 186–196, doi:10.1016/j.commatsci.2019.01.006.
34. Zhu, H.; Liu, F.; Ye, Y.; Chen, L.; Liu, J.; Gui, A.; Zhang, J.; Dong, C. Application of machine learning algorithms in quality assurance of fermentation process of black tea— based on electrical properties. *J. Food Eng.* **2019**, *263*, 165–172, doi:10.1016/j.jfoodeng.2019.06.009.
35. Ge, Z.; Chen, T.; Song, Z. Quality prediction for polypropylene production process based on CLGPR model. *Control. Eng. Pract.* **2011**, *19*, 423–432, doi:10.1016/j.conengprac.2011.01.002.
36. Zheng, W.; Gao, X.; Liu, Y.; Wang, L.; Yang, J.; Gao, Z. Industrial Mooney viscosity prediction using fast semi-supervised empirical model. *Chemom. Intell. Lab. Syst.* **2017**, *171*, 86–92, doi:10.1016/j.chemolab.2017.10.009.
37. Liang, Y.; Liu, Z.; Liu, W. A co-training style semi-supervised artificial neural network modeling and its application in thermal conductivity prediction of polymeric composites filled with BN sheets. *Energy AI* **2021**, *4*, 100052, doi:10.1016/j.egyai.2021.100052.
38. Yan, W.; Guo, P.; Tian, Y.; Gao, J. A Framework and Modeling Method of Data-Driven Soft Sensors Based on Semisupervised Gaussian Regression. *Ind. Eng. Chem. Res.* **2016**, *55*, 7394–7401, doi:10.1021/acs.iecr.5b04118.
39. Liu, Y.; Yang, C.; Gao, Z.; Yao, Y. Ensemble deep kernel learning with application to quality prediction in industrial polymerization processes. *Chemom. Intell. Lab. Syst.* **2018**, *174*, 15–21, doi:10.1016/j.chemolab.2018.01.008.
40. Yin, X.; Niu, Z.; He, Z.; Li, Z.; hee Lee, D. Ensemble deep learning based semi-supervised soft sensor modeling method and its application on quality prediction for coal preparation process. *Adv. Eng. Inform.* **2020**, *46*, 101136, doi:10.1016/j.aei.2020.101136.
41. He, X.; Ji, J.; Liu, K.; Gao, Z.; Liu, Y. Soft sensing of silicon content via bagging local semi-supervised models. *Sensors* **2019**, *19*, 3814, doi:10.3390/s19173814.
42. Ma, W.; Cheng, F.; Xu, Y.; Wen, Q.; Liu, Y. Probabilistic Representation and Inverse Design of Metamaterials Based on a Deep Generative Model with Semi-Supervised Learning Strategy. *Adv. Mater.* **2019**, *31*, 1–9, doi:10.1002/adma.201901111.
43. Okaro, I.A.; Jayasinghe, S.; Sutcliffe, C.; Black, K.; Paoletti, P.; Green, P.L. Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning. *Addit. Manuf.* **2019**, *27*, 42–53, doi:10.1016/j.addma.2019.01.006.
44. Zhang, Y.; Lu, Z. Exploring semi-supervised variational autoencoders for biomedical relation extraction. *Methods* **2019**, *166*, 112–119, doi:10.1016/j.ymeth.2019.02.021.
45. Sahoo, P.; Roy, I.; Wang, Z.; Mi, F.; Yu, L.; Balasubramani, P.; Khan, L.; Stoddart, J.F. MultiCon: A Semi-Supervised Approach for Predicting Drug Function from Chemical Structure Analysis. *J. Chem. Inf. Model.* **2020**, *60*, 5995–6006, doi:10.1021/acs.jcim.0c00801.
46. Yu, H.; Ji, J.; Li, P.; Shao, F.; Wu, S.; Sui, Y.; Li, S.; He, F.; Liu, J. Semi-Supervised Hybrid Local Kernel Regression for Soft Sensor Modelling of Rubber-Mixing Process. *Adv. Polym. Technol.* **2020**, *2020*, 6981302, doi:10.1155/2020/6981302.
47. Zheng, J.; Du, J.; Liang, Y.; Liao, Q.; Li, Z.; Zhang, H.; Wu, Y. Deeppipe: A semi-supervised learning for operating condition recognition of multi-product pipelines. *Process. Saf. Environ. Prot.* **2021**, *150*, 510–521, doi:10.1016/j.psep.2021.04.031.
48. Ma, Y.; Zhu, W.; Benton, M.; Romagnoli, J. Continuous control of a polymerization system with deep reinforcement learning. *J. Process. Control.* **2019**, *75*, 40–47, doi:10.1016/j.jprocont.2018.11.004.
49. Singh, V.; Kodamana, H. Reinforcement learning based control of batch polymerisation processes. *IFAC PapersOnLine* **2020**, *53*, 667–672, doi:10.1016/j.ifacol.2020.06.111.

50. Nian, R.; Liu, J.; Huang, B. A review On reinforcement learning: Introduction and applications in industrial process control. *Comput. Chem. Eng.* **2020**, *139*, 106886, doi:10.1016/j.compchemeng.2020.106886.
51. Venkatasubramanian, V. The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE J.* **2019**, *65*, 466–478, doi:10.1002/aic.16489.
52. Zendehboudi, S.; Rezaei, N.; Lohi, A. Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review. *Appl. Energy* **2018**, *228*, 2539–2566, doi:10.1016/j.apenergy.2018.06.051.
53. Uhlemann, J.; Costa, R. Product Design and Engineering in Chemical Engineering: Past , Present State , and Future. *Chem. Eng. Technol.* **2019**, *42*, 2258–2274, doi:10.1002/ceat.201900236.
54. Meng, Y.; Yu, S.; Zhang, J.; Qin, J.; Dong, Z.; Lu, G. Hybrid modeling based on mechanistic and data-driven approaches for cane sugar crystallization. *J. Food Eng.* **2019**, *257*, 44–55, doi:10.1016/j.jfoodeng.2019.03.026.
55. Li, B.; Lin, Y.; Yu, W.; Wilson, D.I.; Young, B.R. Application of mechanistic modelling and machine learning for cream cheese fermentation pH prediction. *J. Chem. Technol. Biotechnol.* **2021**, *96*, 125–133, doi:10.1002/jctb.6517.
56. Zhang, X.; Zhou, T.; Zhang, L.; Fung, K.Y.; Ng, K.M. Food Product Design: A Hybrid Machine Learning and Mechanistic Modeling Approach. *Ind. Eng. Chem. Res.* **2019**, *58*, 16743–16752, doi:10.1021/acs.iecr.9b02462.
57. von Stosch, M.; Davy, S.; Francois, K.; Galvanauskas, V.; Hamelink, J.M.; Luebbert, A.; Mayer, M.; Oliveira, R.; O'Kennedy, R.; Rice, P.; et al. Hybrid modeling for quality by design and PAT-benefits and challenges of applications in biopharmaceutical industry. *Biotechnol. J.* **2014**, *9*, 719–726, doi:10.1002/biot.201300385.
58. Drăgoi, E.N.; Curteanu, S.; Fissore, D.; Curteanu, S.; Fissore, D. On the Use of Artificial Neural Networks to Monitor a Pharmaceutical Freeze-Drying Process On the Use of Artificial Neural Networks to Monitor a Pharmaceutical Freeze-Drying Process. *Dry. Technol.* **2013**, *31*, 72–81, doi:10.1080/07373937.2012.718308.
59. Calvo, F.; Gómez, J.M.; Ricardez-Sandoval, L.; Alvarez, O. Integrated design of emulsified cosmetic products: A review. *Chem. Eng. Res. Des.* **2020**, *161*, 279–303, doi:10.1016/j.cherd.2020.07.014.
60. Sadowski, P.; Fooshee, D.; Subrahmanya, N.; Baldi, P. Synergies between quantum mechanics and machine learning in reaction prediction. *J. Chem. Inf. Model.* **2016**, *56*, 2125–2128, doi:10.1021/acs.jcim.6b00351.
61. Castéran, F.; Ibanez, R.; Argerich, C.; Delage, K.; Chinesta, F.; Cassagnau, P. Application of Machine Learning Tools for the Improvement of Reactive Extrusion Simulation. *Macromol. Mater. Eng.* **2020**, *305*, 2000375, doi:10.1002/mame.202000375.
62. Ghiba, L.; Drăgoi, E.N.; Curteanu, S. Neural network-based hybrid models developed for free radical polymerization of styrene. *Polym. Eng. Sci.* **2021**, *61*, 716–730, doi:10.1002/pen.25611.
63. Curteanu, S.; Leon, F. Hybrid neural network models applied to a free radical polymerization process. *Polym. Plast. Technol. Eng.* **2006**, *45*, 1013–1023, doi:10.1080/03602550600726285.
64. Ng, C.W.; Hussain, M.A. Hybrid neural network-prior knowledge model in temperature control of a semi-batch polymerization process. *Chem. Eng. Process. Process. Intensif.* **2004**, *43*, 559–570, doi:10.1016/S0255-2701(03)00109-0.
65. Qi, C.; Ly, H.B.; Chen, Q.; Le, T.T.; Le, V.M.; Pham, B.T. Flocculation-dewatering prediction of fine mineral tailings using a hybrid machine learning approach. *Chemosphere* **2020**, *244*, 125450, doi:10.1016/j.chemosphere.2019.125450.
66. Bi, K.; Qiu, T.; Huang, Y. A deep learning method for yogurt preferences prediction using sensory attributes. *Processes* **2020**, *8*, 518, doi:10.3390/PR8050518.
67. Chen, L.; Kim, C.; Batra, R.; Lightstone, J.P.; Wu, C.; Li, Z.; Deshmukh, A.A.; Wang, Y.; Tran, H.D.; Vashishta, P.; Sotzing, G.A.; Cao, Y.; Ramprasad, R. Frequency-dependent dielectric constant prediction of polymers using machine learning. *NPJ Comput. Mater.* **2020**, *6*, 30–32, doi:10.1038/s41524-020-0333-6.
68. Batra, R.; Dai, H.; Huan, T.D.; Chen, L.; Kim, C.; Gutekunst, W.R.; Song, L.; Ramprasad, R. Polymers for Extreme Conditions Designed Using Syntax-Directed Variational Autoencoders. *Chem. Mater.* **2020**, *32*, 10489–10500, doi:10.1021/acs.chemmater.0c03332.
69. Zhou, T.; Gani, R.; Sundmacher, K. Hybrid data-driven and mechanistic modeling approaches for multiscale material and process design. *Engineering* **2021**, doi:10.1016/j.eng.2020.12.022.
70. McBride, K.; Sanchez Medina, E.I.; Sundmacher, K. Hybrid Semi-parametric Modeling in Separation Processes: A Review. *Chem. Ingenieur-Technik* **2020**, *92*, 842–855, doi:10.1002/cite.202000025.
71. Zhang, X.; Ding, X.; Song, Z.; Zhou, T.; Sundmacher, K. Integrated ionic liquid and rate-based absorption process design for gas separation: Global optimization using hybrid models. *AIChE J.* **2021**, e17340, doi:10.1002/aic.17340.
72. Zhang, L.; Mao, H.; Liu, Q.; Gani, R. Chemical product design–recent advances and perspectives. *Curr. Opin. Chem. Eng.* **2020**, *27*, 22–34, doi:10.1016/j.coche.2019.10.005.
73. Costa, R.; Moggridge, G.D.; Saraiva, P.M. Chemical Product Engineering: An Emerging Paradigm within Chemical Engineering. *Aiche J.* **2006**, *52*, 1976–1986, doi:10.1002/aic.
74. Arrieta-Escobar, J.A.; Camargo, M.; Morel, L.; Orjuela, A. Current approaches on chemical product design: A study of opportunities identification for integrated methodologies. In Proceedings of the Towards the Digital World and Industry X.0-Proceedings of the 29th International Conference of the International Association for Management of Technology, IAMOT 2020, Cairo, Egypt, 13–17 September 2020; pp. 785–794.
75. Ng, K.M.; Gani, R. Chemical product design: Advances in and proposed directions for research and teaching. *Comput. Chem. Eng.* **2019**. *126*, 147–156, doi:10.1016/j.compchemeng.2019.04.008.
76. Cussler, E.L. *Chemical Product Design*; Cambridge University Press: Cambridge, UK; New York, NY, USA, 2011.
77. Hill, M. Chemical Product Engineering-The third paradigm. *Comput. Chem. Eng.* **2009**, *33*, 947–953, doi:10.1016/j.compchemeng.2008.11.013.

78. Taifouris, M.; Martín, M.; Martínez, A.; Esquejo, N. Challenges in the design of formulated products: multiscale process and product design. *Curr. Opin. Chem. Eng.* **2020**, *27*, 1–9, doi:10.1016/j.coche.2019.10.001.
79. Fischer, A. Artificial Intelligence Colloquium: Accelerating Chemistry with AI. 2019. Available online: https://theengineeringofconsciousexperience.com/artificial-intelligence-colloquium-accelerating-chemistry-with-ai/ (accessed on 10 August 2021).
80. Goussard, V.; Duprat, F.; Ploix, J.L.; Dreyfus, G.; Nardello-Rataj, V.; Aubry, J.M. A New Machine-Learning Tool for Fast Estimation of Liquid Viscosity. Application to Cosmetic Oils. *J. Chem. Inf. Model.* **2020**, *60*, 2012–2023, doi:10.1021/acs.jcim.0c00083.
81. Dobbelaere, M.R.; Plehiers, P.P.; Van de Vijver, R.; Stevens, C.V.; Van Geem, K.M. Learning Molecular Representations for Thermochemistry Prediction of Cyclic Hydrocarbons and Oxygenates. *J. Phys. Chem. A* **2021**, *125*, 5166–5179, doi:10.1021/acs.jpca.1c01956.
82. Lo, Y.C.; Rensi, S.E.; Torng, W.; Altman, R.B. Machine learning in chemoinformatics and drug discovery. *Drug Discov. Today* **2018**, *23*, 1538–1546, doi:10.1016/j.drudis.2018.05.010.
83. Smith, J.S.; Roitberg, A.E.; Isayev, O. Transforming Computational Drug Discovery with Machine Learning and AI. *ACS Med. Chem. Lett.* **2018**, *9*, 1065–1069, doi:10.1021/acsmedchemlett.8b00437.
84. David, L.; Thakkar, A.; Mercado, R.; Engkvist, O. Molecular representations in AI-driven drug discovery: A review and practical guide. *J. Cheminform.* **2020**, *12*, 56, doi:10.1186/s13321-020-00460-5.
85. Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **2018**, *361*, 360–365, doi:10.1126/science.aat2663.
86. Staker, J.; Marques, G.; Dakka, J. Chapter 15 Representation Learning in Chemistry. In *Machine Learning in Chemistry: The Impact of Artificial Intelligence*; The Royal Society of Chemistry: London, UK, 2020; pp. 372–397. doi:10.1039/9781839160233-00372.
87. Alshehri, A.S.; Gani, R.; You, F. Deep learning and knowle dge-base d methods for computer-aided molecular design—Toward a unified approach: State-of-the-art and future directions. *Comput. Chem. Eng.* **2020**, *141*, 107005, doi:10.1016/j.compchemeng.2020.107005.
88. Audus, D.J.; De Pablo, J.J. Polymer Informatics: Opportunities and Challenges. *ACS Macro Lett.* **2017**, *6*, 1078–1082, doi:10.1021/acsmacrolett.7b00228.
89. Wei, J.N. Exploring Machine Learning Applications to Enable Next-Generation Chemistry. Ph.D. Thesis, Harvard University, Cambridge, MA, USA, 2019.
90. Luan, F. Classification of the fragrance properties of chemical compounds based on support vector machine and linear discriminant analysis. *Flavour Fragr. J.* **2007**, *23*, 311–316, doi:10.1002/ffj.
91. Kennedy, K.; Cal, R.; Casey, R.; Lopez, C.; Adelfio, A.; Molloy, B.; Wall, A.M.; Holton, T.A.; Khaldi, N. The anti-ageing effects of a natural peptide discovered by artificial intelligence. *Int. J. Cosmet. Sci.* **2020**, *42*, 388–398, doi:10.1111/ics.12635.
92. Lightstone, J.P.; Chen, L.; Kim, C.; Batra, R.; Ramprasad, R. Refractive index prediction models for polymers using machine learning. *J. Appl. Phys.* **2020**, *127*, 215105, doi:10.1063/5.0008026.
93. Zhang, Y.; Xu, X. Machine learning glass transition temperature of polymers. *Heliyon* **2020**, *6*, e05055, doi:10.1016/j.heliyon.2020.e05055.
94. Yang, Y.; Zhang, X.; Yin, J.; Yu, X. Rapid and Nondestructive On-Site Classification Method for Consumer-Grade Plastics Based on Portable NIR Spectrometer and Machine Learning. *J. Spectrosc.* **2020**, *2020*, 6631234, doi:10.1155/2020/6631234.
95. Bieler, M.; Reutlinger, M.; Rodrigues, T.; Schneider, P.; Kriegl, J.M.; Schneider, G. Designing Multi-target Compound Libraries with Gaussian Process Models. *Mol. Inform.* **2016**, *35*, 192–198. doi:10.1002/minf.201501012.
96. Zhu, G.; Kim, C.; Chandrasekarn, A.; Everett, J.; Ramprasad, R.; Lively, R.P. Polymer genome-based prediction of gas permeabilities in polymers. *J. Polym. Eng.* **2020**, *40*, 451–457, doi:10.1515/polyeng-2019-0329.
97. Song, Z.; Shi, H.; Zhang, X.; Zhou, T. Prediction of CO_2 solubility in ionic liquids using machine learning methods. *Chem. Eng. Sci.* **2020**, *223*, 115752, doi:10.1016/j.ces.2020.115752.
98. Doan Tran, H.; Kim, C.; Chen, L.; Chandrasekaran, A.; Batra, R.; Venkatram, S.; Kamal, D.; Lightstone, J.P.; Gurnani, R.; Shetty, P.; et al. Machine-learning predictions of polymer properties with Polymer Genome. *J. Appl. Phys.* **2020**, *128*, 171104, doi:10.1063/5.0023759.
99. Zhang, X.; Wang, J.; Song, Z.; Zhou, T. Data-Driven Ionic Liquid Design for CO_2 Capture: Molecular Structure Optimization and DFT Veri fi cation. *Ind. Eng. Chem. Res.* **2021**, *60*, 9992–10000, doi:10.1021/acs.iecr.1c01384.
100. Haghighatlari, M.; Hachmann, J. Advances of machine learning in molecular modeling and simulation. *Curr. Opin. Chem. Eng.* **2019**, *23*, 51–57, doi:10.1016/j.coche.2019.02.009.
101. Pilania, G.; Wang, C.; Jiang, X.; Rajasekaran, S.; Ramprasad, R. Accelerating materials property predictions using machine learning. *Sci. Rep.* **2013**, *3*, 2810, doi:10.1038/srep02810.
102. Yeo, C.S.H.; Xie, Q.; Wang, X.; Zhang, S. Understanding and optimization of thin film nanocomposite membranes for reverse osmosis with machine learning. *J. Membr. Sci.* **2020**, *606*, 118135, doi:10.1016/j.memsci.2020.118135.
103. Yan, Y.; Mattisson, T.; Moldenhauer, P.; Anthony, E.J.; Clough, P.T. Applying machine learning algorithms in estimating the performance of heterogeneous, multi-component materials as oxygen carriers for chemical-looping processes. *Chem. Eng. J.* **2020**, *387*, 124072, doi:10.1016/j.cej.2020.124072.
104. Sun, Y.; Hewitt, M.; Wilkinson, S.C.; Davey, N.; Adams, R.G.; Gullick, D.R.; Moss, G.P. Development of a Gaussian Process–feature selection model to characterise (poly)dimethylsiloxane (Silastic®) membrane permeation. *J. Pharm. Pharmacol.* **2020**, *72*, 873–888, doi:10.1111/jphp.13263.
105. Ju, S.; Shimizu, S.; Shiomi, J. Designing thermal functional materials by coupling thermal transport calculations and machine learning. *J. Appl. Phys.* **2020**, *128*, 161102, doi:10.1063/5.0017042.

106. Jiao, P.; Alavi, A.H. Artificial intelligence-enabled smart mechanical metamaterials: advent and future trends. *Int. Mater. Rev.* **2021**, *66*, 365–393, doi:10.1080/09506608.2020.1815394.
107. Chen, C.; Zuo, Y.; Ye, W.; Li, X.; Deng, Z.; Ong, S.P. A Critical Review of Machine Learning of Energy Materials. *Adv. Energy Mater.* **2020**, *10*, 1903242, doi:10.1002/aenm.201903242.
108. Christensen, T.; Loh, C.; Picek, S.; Jakobović, D.; Jing, L.; Fisher, S.; Ceperic, V.; Joannopoulos, J.D.; Soljačić, M. Predictive and generative machine learning models for photonic crystals. *Nanophotonics* **2020**, *9*, 4183–4192, doi:10.1515/nanoph-2020-0197.
109. Xie, Y.; Zhang, C.; Hu, X.; Zhang, C.; Kelley, S.P.; Atwood, J.L.; Lin, J. Machine Learning Assisted Synthesis of Metal-Organic Nanocapsules. *J. Am. Chem. Soc.* **2020**, *142*, 1475–1481, doi:10.1021/jacs.9b11569.
110. Li, F.; Han, J.; Cao, T.; Lam, W.; Fan, B.; Tang, W.; Chen, S.; Fok, K.L.; Li, L. Design of self-assembly dipeptide hydrogels and machine learning via their chemical features. *Proc. Natl. Acad. Sci. USA* **2019**, *166*, 11259–11264, doi:10.1073/pnas.1903376116.
111. Gu, G.H.; Noh, J.; Kim, I.; Jung, Y. Machine learning for renewable energy materials. *J. Mater. Chem. A* **2019**, *7*, 17096–17117, doi:10.1039/c9ta02356a.
112. Conduit, B.D.; Illston, T.; Baker, S.; Duggappa, D.V.; Harding, S.; Stone, H.J.; Conduit, G.J. Probabilistic neural network identification of an alloy for direct laser deposition. *Mater. Des.* **2019**, *168*, 107644, doi:10.1016/j.matdes.2019.107644.
113. Balachandran, P.V. Machine learning guided design of functional materials with targeted properties. *Comput. Mater. Sci.* **2019**, *164*, 82–90, doi:10.1016/j.commatsci.2019.03.057.
114. Noack, M.M.; Doerk, G.S.; Li, R.; Streit, J.K.; Vaia, R.A.; Yager, K.G.; Fukuto, M. Autonomous materials discovery driven by Gaussian process regression with inhomogeneous measurement noise and anisotropic kernels. *Sci. Rep.* **2020**, *10*, 17663, doi:10.1038/s41598-020-74394-1.
115. Mansouri Tehrani, A.; Oliynyk, A.O.; Parry, M.; Rizvi, Z.; Couper, S.; Lin, F.; Miyagi, L.; Sparks, T.D.; Brgoch, J. Machine Learning Directed Search for Ultraincompressible, Superhard Materials. *J. Am. Chem. Soc.* **2018**, *140*, 9844–9853, doi:10.1021/jacs.8b02717.
116. Tabor, D.P.; Roch, L.M.; Saikin, S.K.; Kreisbeck, C.; Sheberla, D.; Montoya, J.H.; Dwaraknath, S.; Aykol, M.; Ortiz, C.; Tribukait, H.; et al. Accelerating the discovery of materials for clean energy in the era of smart automation. *Nat. Rev. Mater.* **2018**, *3*, 5–20, doi:10.1038/s41578-018-0005-z.
117. Saeki, A. Evaluation-oriented exploration of photo energy conversion systems: from fundamental optoelectronics and material screening to the combination with data science. *Polym. J.* **2020**, *52*, 1307–1321, doi:10.1038/s41428-020-00399-2.
118. Kayala, M.A.; Baldi, P. ReactionPredictor: Prediction of complex chemical reactions at the mechanistic level using machine learning. *J. Chem. Inf. Model.* **2012**, *52*, 2526–2540, doi:10.1021/ci3003039.
119. Schwaller, P.; Gaudin, T.; Lányi, D.; Bekas, C.; Laino, T. "Found in Translation": Predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chem. Sci.* **2018**, *9*, 6091–6098, doi:10.1039/c8sc02339e.
120. Kayala, M.A.; Azencott, C.A.; Chen, J.H.; Baldi, P. Learning to predict chemical reactions. *J. Chem. Inf. Model.* **2011**, *51*, 2209–2222, doi:10.1021/ci200207y.
121. Coley, C.W.; Barzilay, R.; Jaakkola, T.S.; Green, W.H.; Jensen, K.F. Prediction of Organic Reaction Outcomes Using Machine Learning. *ACS Cent. Sci.* **2017**, *3*, 434–443, doi:10.1021/acscentsci.7b00064.
122. Segler, M.H.; Waller, M.P. Neural-Symbolic Machine Learning for Retrosynthesis and Reaction Prediction. *Chem. Eur. J.* **2017**, *23*, 5966–5971, doi:10.1002/chem.201605499.
123. Segler, M.H.; Waller, M.P. Modelling Chemical Reasoning to Predict and Invent Reactions. *Chem. Eur. J.* **2017**, *23*, 6118–6128, doi:10.1002/chem.201604556.
124. Segler, M.H.; Preuss, M.; Waller, M.P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **2018**, *555*, 604–610, doi:10.1038/nature25978.
125. Schwaller, P.; Laino, T.; Gaudin, T.; Bolgar, P.; Hunter, C.A.; Bekas, C.; Lee, A.A. Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS Cent. Sci.* **2019**, *5*, 1572–1583, doi:10.1021/acscentsci.9b00576.
126. Schwaller, P.; Petraglia, R.; Zullo, V.; Nair, V.H.; Haeuselmann, R.A.; Pisoni, R.; Bekas, C.; Iuliano, A.; Laino, T. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chem. Sci.* **2020**, *11*, 3316–3325, doi:10.1039/c9sc05704h.
127. Schwaller, P.; Hoover, B.; Reymond, J.L.; Strobelt, H.; Laino, T. Extraction of organic chemistry grammar from unsupervised learning of chemical reactions. *Sci. Adv.* **2021**, *7*, eabe4166, doi:10.1126/SCIADV.ABE4166.
128. Gao, H.; Struble, T.J.; Coley, C.W.; Wang, Y.; Green, W.H.; Jensen, K.F. Using Machine Learning to Predict Suitable Conditions for Organic Reactions. *ACS Cent. Sci.* **2018**, *4*, 1465–1476, doi:10.1021/acscentsci.8b00357.
129. Wei, J.N.; Duvenaud, D.; Aspuru-Guzik, A. Neural networks for the prediction of organic chemistry reactions. *ACS Cent. Sci.* **2016**, *2*, 725–732, doi:10.1021/acscentsci.6b00219.
130. Ishida, S.; Terayama, K.; Kojima, R.; Takasu, K.; Okuno, Y. Prediction and Interpretable Visualization of Retrosynthetic Reactions Using Graph Convolutional Networks. *J. Chem. Inf. Model.* **2019**, *59*, 5026–5033, doi:10.1021/acs.jcim.9b00538.
131. Nam, J.; Kim, J. Linking the Neural Machine Translation and the Prediction of Organic Chemistry Reactions. *arXiv* **2016**, arXiv:1612.09529.
132. McCoy, J.T.; Auret, L. Machine learning applications in minerals processing: A review. *Miner. Eng.* **2019**, *132*, 95–109, doi:10.1016/j.mineng.2018.12.004.
133. Curteanu, S.; Leon, F.; Mircea-Vicoveanu, A.M.; Logofătu, D. Regression methods based on nearest neighbors with adaptive distance metrics applied to a polymerization process. *Mathematics* **2021**, *9*, 547, doi:10.3390/math9050547.

134. Curteanu, S. Chapter 10 Machine Learning Techniques Applied to a Complex Polymerization Process. In *Machine Learning in Chemistry: The Impact of Artificial Intelligence*; The Royal Society of Chemistry: London, UK, 2020; pp. 227–250. doi:10.1039/9781839160233-00227.
135. Meimaroglou, D.; Florez, D.; Hu, G.H. A kinetic modeling framework for the peroxide-initiated radical polymerization of styrene in the presence of rubber particles from recycled tires. *Chem. Eng. Sci.* **2020**, under review.
136. Khayyam, H.; Jazar, R.N.; Nunna, S.; Golkarnarenji, G.; Badii, K.; Fakhrhoseini, S.M.; Kumar, S.; Naebe, M. PAN precursor fabrication, applications and thermal stabilization process in carbon fiber production: Experimental and mathematical modelling. *Prog. Mater. Sci.* **2020**, *107*, 100575, doi:10.1016/j.pmatsci.2019.100575.
137. Kramer, A.; Morgado-Dias, F. Artificial intelligence in process control applications and energy saving: A review and outlook. *Greenh. Gases Sci. Technol.* **2020**, *10*, 1133–1150, doi:10.1002/ghg.1962.
138. Dong, E.L.; Song, J.H.; Song, S.O.; En, S.Y. Weighted support vector machine for quality estimation in the polymerization process. *Ind. Eng. Chem. Res.* **2005**, *44*, 2101–2105, doi:10.1021/ie049908e.
139. Zhao, P.; Zhang, J.; Dong, Z.; Huang, J.; Zhou, H.; Fu, J.; Turng, L.S. Intelligent Injection Molding on Sensing, Optimization, and Control. *Adv. Polym. Technol.* **2020**, *2020*, 7023616. doi:10.1155/2020/7023616.
140. Khan, A.; Shamsi, M.H.; Choi, T.S. Correlating dynamical mechanical properties with temperature and clay composition of polymer-clay nanocomposites. *Comput. Mater. Sci.* **2009**, *45*, 257–265, doi:10.1016/j.commatsci.2008.09.027.
141. Li, C.; Rubín De Celis Leal, D.; Rana, S.; Gupta, S.; Sutti, A.; Greenhill, S.; Slezak, T.; Height, M.; Venkatesh, S. Rapid Bayesian optimisation for synthesis of short polymer fiber materials. *Sci. Rep.* **2017**, *7*, 5683, doi:10.1038/s41598-017-05723-0.
142. Ibañez, R.; Casteran, F.; Argerich, C.; Ghnatios, C.; Hascoet, N.; Ammar, A.; Cassagnau, P.; Chinesta, F. On the data-driven modeling of reactive extrusion. *Fluids* **2020**, *5*, 94, doi:10.3390/fluids5020094.
143. Curteanu, S.; Leon, F.; Galea, D. Neural network models for free radical polymerization of methyl methacrylate Neural Network Models for Free Radical Polymerization of Methyl Methacrylate. *Eurasian Chemtech J.* **2003**, *5*, 225–231.
144. Curteanu, S. Direct and inverse neural network modeling in free radical polymerization. *Cent. Eur. J. Chem.* **2004**, *2*, 113–140, doi:10.2478/BF02476187.
145. Rodríguez-Dorado, R.; Landín, M.; Altai, A.; Russo, P.; Aquino, R.P.; Del Gaudio, P. A novel method for the production of core-shell microparticles by inverse gelation optimized with artificial intelligent tools. *Int. J. Pharm.* **2018**, *538*, 97–104, doi:10.1016/j.ijpharm.2018.01.023.
146. Rouco, H.; Diaz-Rodriguez, P.; Rama-Molinos, S.; Remuñán-López, C.; Landin, M. Delimiting the knowledge space and the design space of nanostructured lipid carriers through Artificial Intelligence tools. *Int. J. Pharm.* **2018**, *553*, 522–530, doi:10.1016/j.ijpharm.2018.10.058.
147. Wang, K.; Han, L.; Mustakis, J.; Li, B.; Magano, J.; Damon, D.B.; Dion, A.; Maloney, M.T.; Post, R.; Li, R. Kinetic and Data-Driven Reaction Analysis for Pharmaceutical Process Development. *Ind. Eng. Chem. Res.* **2020**, *59*, 2409–2421, doi:10.1021/acs.iecr.9b03578.
148. Sarmadi, M.; Behrens, A.M.; McHugh, K.J.; Contreras, H.T.; Tochka, Z.L.; Lu, X.; Langer, R.; Jaklenec, A. Modeling, design, and machine learning-based framework for optimal injectability of microparticle-based drug formulations. *Sci. Adv.* **2020**, *6*, abb6594, doi:10.1126/sciadv.abb6594.
149. Jhamb, S.; Enekvist, M.; Liang, X.; Zhang, X.; Dam-Johansen, K.; Kontogeorgis, G.M. A review of computer-aided design of paints and coatings. *Curr. Opin. Chem. Eng.* **2020**, *27*, 107–120, doi:10.1016/j.coche.2019.12.005.
150. Jasso-Salcedo, A.B.; Hoppe, S.; Pla, F.; Escobar-Barrios, V.A.; Camargo, M.; Meimaroglou, D. Modeling and optimization of a photocatalytic process: Degradation of endocrine disruptor compounds by Ag/ZnO. *Chem. Eng. Res. Des.* **2017**, *128*, 174–191, doi:10.1016/j.cherd.2017.10.012.
151. Jeguirim, S.E.G.; Dhouib, A.B.; Sahnoun, M.; Cheikhrouhou, M.; Schacher, L.; Adolphe, D. The use of fuzzy logic and neural networks models for sensory properties prediction from process and structure parameters of knitted fabrics. *J. Intell. Manuf.* **2011**, *22*, 873–884, doi:10.1007/s10845-009-0362-y.
152. Golkarnarenji, G.; Naebe, M.; Badii, K.; Milani, A.S.; Jazar, R.N.; Khayyam, H. A machine learning case study with limited data for prediction of carbon fiber mechanical properties. *Comput. Ind.* **2019**, *105*, 123–132, doi:10.1016/j.compind.2018.11.004.
153. Wang, Q.; Lonergan, S.M.; Yu, C. Rapid determination of pork sensory quality using Raman spectroscopy. *Meat Sci.* **2012**, *91*, 232–239, doi:10.1016/j.meatsci.2012.01.017.
154. Ruan, D. *Intelligent Sensory Evaluation: Methodologies and Applications*; Springer Berlin Heidelberg: Berlin/Heidelberg, Germany, 2004.
155. Zeng, X.; Ruan, D.; Koehl, L. Intelligent sensory evaluation: Concepts, implementations, and applications. *Math. Comput. Simul.* **2008**, *77*, 443–452, doi:10.1016/j.matcom.2007.11.013.
156. Ouyang, Q.; Chen, Q.; Zhao, J. Intelligent sensing sensory quality of Chinese rice wine using near infrared spectroscopy and nonlinear tools. *Spectrochim. Acta Part Mol. Biomol. Spectrosc.* **2016**, *154*, 42–46, doi:10.1016/j.saa.2015.10.011.
157. Gunaratne, T.M.; Viejo, C.G.; Gunaratne, N.M.; Torrico, D.D.; Dunshea, F.R.; Fuentes, S. Chocolate quality assessment based on chemical fingerprinting using near infra-red and machine learning modeling. *Foods* **2019**, *8*, 426, doi:10.3390/foods8100426.
158. Sanahuja, S.; Fédou, M.; Briesen, H. Classification of puffed snacks freshness based on crispiness-related mechanical and acoustical properties. *J. Food Eng.* **2018**, *226*, 53–64, doi:10.1016/j.jfoodeng.2017.12.013.
159. Bahamonde, A.; Díez, J.; Quevedo, J.R.; Luaces, O.; del Coz, J.J. How to learn consumer preferences from the analysis of sensory data by means of support vector machines (SVM). *Trends Food Sci. Technol.* **2007**, *18*, 20–28, doi:10.1016/j.tifs.2006.07.014.

160. Zhi, R.; Zhao, L.; Shi, J. Improving the sensory quality of flavored liquid milk by engaging sensory analysis and consumer preference. *J. Dairy Sci.* **2016**, *99*, 5305–5317, doi:10.3168/jds.2015-10612.
161. Krishnamurthy, R.; Srivastava, A.K.; Paton, J.E.; Bell, G.A.; Levy, D.C. Prediction of consumer liking from trained sensory panel information: Evaluation of neural networks. *Food Qual. Prefer.* **2007**, *18*, 275–285, doi:10.1016/j.foodqual.2006.01.001.
162. Rocha, R.S.; Calvalcanti, R.N.; Silva, R.; Guimarães, J.T.; Balthazar, C.F.; Pimentel, T.C.; Esmerino, E.A.; Freitas, M.Q.; Granato, D.; Costa, R.G.; et al. Consumer acceptance and sensory drivers of liking of Minas Frescal Minas cheese manufactured using milk subjected to ohmic heating: Performance of machine learning methods. *LWT* **2020**, *126*, 109342, doi:10.1016/j.lwt.2020.109342.
163. Fuentes, S.; Torrico, D.D.; Tongson, E.; Viejo, C.G. Machine learning modeling of wine sensory profiles and color of vertical vintages of pinot noir based on chemical fingerprinting, weather and management data. *Sensors* **2020**, *20*, 3618, doi:10.3390/s20133618.
164. Liu, P.; Zhu, X.; Hu, X.; Xiong, A.; Wen, J.; Li, H.; Ai, S.; Wu, R. Local tangent space alignment and relevance vector machine as nonlinear methods for estimating sensory quality of tea using NIR spectroscopy. *Vib. Spectrosc.* **2019**, *103*, 102923, doi:10.1016/j.vibspec.2019.05.005.
165. Vigneau, E.; Courcoux, P.; Symoneaux, R.; Guérin, L.; Villière, A. Random forests: A machine learning methodology to highlight the volatile organic compounds involved in olfactory perception. *Food Qual. Prefer.* **2018**, *68*, 135–145, doi:10.1016/j.foodqual.2018.02.008.
166. Viejo, C.G.; Fuentes, S. A Digital Approach to Model Quality and Sensory Traits of Beers Fermented under Sonication Based on Chemical Fingerprinting. *Fermentation* **2020**, *6*, 73, doi:10.3390/FERMENTATION6030073.
167. Nozaki, Y.; Nakamoto, T. Correction: Predictive modeling for odor character of a chemical using machine learning combined with natural language processing (PLoS ONE (2018) 13, 6 (e0198475) DOI: 10.1371/journal.pone.0198475). *PLoS ONE* **2018**, *13*, e0208962. doi:10.1371/journal.pone.0208962.
168. Zhang, X.; Zhou, T.; Ng, K.M. Optimization-based cosmetic formulation: Integration of mechanistic model, surrogate model, and heuristics. *AIChE J.* **2021**, *67*, 1–16, doi:10.1002/aic.17064.
169. Gonzalez Viejo, C.; Fuentes, S.; Torrico, D.; Lee, M.; Hu, Y.; Chakraborty, S.; Dunshea, F. The Effect of Soundwaves on Foamability Properties and Sensory of Beers with a Machine Learning Modeling Approach. *Beverages* **2018**, *4*, 53, doi:10.3390/beverages4030053.
170. Lerma-García, M.J.; Cerretani, L.; Cevoli, C.; Simó-Alfonso, E.F.; Bendini, A.; Toschi, T.G. Use of electronic nose to determine defect percentage in oils. Comparison with sensory panel results. *Sens. Actuators B Chem.* **2010**, *147*, 283–289, doi:10.1016/j.snb.2010.03.058.
171. Goodfellow, I. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016.
172. Martynenko, A.; Misra, N.N. Machine learning in drying. *Dry. Technol.* **2020**, *38*, 596–609, doi:10.1080/07373937.2019.1690502.
173. Lu, N.V.; Tansuchat, R.; Yuizono, T.; Huynh, V.N. Incorporating active learning into machine learning techniques for sensory evaluation of food. *Int. J. Comput. Intell. Syst.* **2020**, *13*, 655–662, doi:10.2991/ijcis.d.200525.001.
174. Al-Jamimi, H.A.; Al-Azani, S.; Saleh, T.A. Supervised machine learning techniques in the desulfurization of oil products for environmental protection: A review. *Process. Saf. Environ. Prot.* **2018**, *120*, 57–71, doi:10.1016/j.psep.2018.08.021.
175. Azencott, C.A. *Introduction au Machine Learning*; Dunod: Malakoff, Malaysia, 2018.
176. Asante-Okyere, S.; Shen, C.; Ziggah, Y.Y.; Rulegeya, M.M.; Zhu, X. Investigating the predictive performance of Gaussian process regression in evaluating reservoir porosity and permeability. *Energies* **2018**, *11*, 3261, doi:10.3390/en11123261.
177. Gong, X.; Yabansu, Y.C.; Collins, P.C.; Kalidindi, S.R. Evaluation of Ti – Mn Alloys for Additive Assays and Gaussian Process Regression. *Materials* **2020**, *13*, 4641.
178. Zhao, P.; Wang, S.; Ying, J.; Fu, J. Non-destructive measurement of cavity pressure during injection molding process based on ultrasonic technology and Gaussian process. *Polym. Test.* **2013**, *32*, 1436–1444, doi:10.1016/j.polymertesting.2013.09.006.
179. Liu, Y.; Gao, Z. Real-time property prediction for an industrial rubber-mixing process with probabilistic ensemble Gaussian process regression models. *J. Appl. Polym. Sci.* **2015**, *132*, 1–9. doi:10.1002/app.41432.
180. Liu, H.; Ong, Y.S.; Shen, X.; Cai, J. When Gaussian Process Meets Big Data: A Review of Scalable GPs. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 4405–4423, doi:10.1109/TNNLS.2019.2957109.
181. Rasmussen, C. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
182. Burkov, A. *Machine Learning Engineering*; True Positive, Inc.: Québec, QC, Canada, 2020.
183. Cartwright, H.M. Chapter 5 Machine Learning in Science – A Role for Mechanical Sympathy? In *Machine Learning in Chemistry: The Impact of Artificial Intelligence*; The Royal Society of Chemistry: London, UK, 2020; pp. 109–135. doi:10.1039/9781839160233-00109.
184. Irwin, B.W.; Levell, J.R.; Whitehead, T.M.; Segall, M.D.; Conduit, G.J. Practical Applications of Deep Learning to Impute Heterogeneous Drug Discovery Data. *J. Chem. Inf. Model.* **2020**, *60*, 2848–2857, doi:10.1021/acs.jcim.0c00443.
185. Whitehead, T.M.; Irwin, B.W.; Hunt, P.; Segall, M.D.; Conduit, G.J. Imputation of Assay Bioactivity Data Using Deep Learning. *J. Chem. Inf. Model.* **2019**, *59*, 1197–1204, doi:10.1021/acs.jcim.8b00768.
186. Stukenbroeker, T.; Clausen, J. Chapter 6 A Prediction of Future States: AI-powered Chemical Innovation for Defense Applications. In *Machine Learning in Chemistry: The Impact of Artificial Intelligence*; The Royal Society of Chemistry: London, UK, 2020; pp. 136–168. doi:10.1039/9781839160233-00136.
187. Sha, W.; Li, Y.; Tang, S.; Tian, J.; Zhao, Y.; Guo, Y.; Zhang, W.; Zhang, X.; Lu, S.; Cao, Y.; et al. Machine learning in polymer informatics. *InfoMat* **2021**, *3*, 353–361, doi:10.1002/inf2.12167.

188. Cai, J.; Chu, X.; Xu, K.; Li, H.; Wei, J. Machine learning-driven new material discovery. *Nanoscale Adv.* **2020**, *2*, 3115–3130, doi:10.1039/d0na00388c.
189. Luo, L.; Yao, Y.; Gao, F.; Zhao, C. Mixed-effects Gaussian process modeling approach with application in injection molding processes. *J. Process. Control.* **2017**, *62*, 37–43,
190. Haghighatlari, M.; Li, J.; Heidar-Zadeh, F.; Liu, Y.; Guan, X.; Head-Gordon, T. Learning to Make Chemical Predictions: The Interplay of Feature Representation, Data, and Machine Learning Methods. *Chem* **2020**, *6*, 1527–1542, doi:10.1016/j.chempr.2020.05.014.
191. Yamada, H.; Liu, C.; Wu, S.; Koyama, Y.; Ju, S.; Shiomi, J.; Morikawa, J.; Yoshida, R. Predicting Materials Properties with Little Data Using Shotgun Transfer Learning. *ACS Cent. Sci.* **2019**, *5*, 1717–1730, doi:10.1021/acscentsci.9b00804.
192. Geiger, A.C.; Cao, Z.; Song, Z.; Ulcickas, J.R.W.; Simpson, G.J. Chapter 18 Autonomous Science: Big Data Tools for Small Data Problems in Chemistry. In *Machine Learning in Chemistry: The Impact of Artificial Intelligence*; The Royal Society of Chemistry: London, UK, 2020; pp. 450–487. doi:10.1039/9781839160233-00450.
193. Stein, H.S.; Gregoire, J.M. Progress and prospects for accelerating materials science with automated and autonomous workflows. *Chem. Sci.* **2019**, *10*, 9640–9649, doi:10.1039/c9sc03766g.
194. Mitchell B.O., J.B. Machine learning methods in chemoinformatics. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2014**, *4*, 468–481, doi:10.1002/wcms.1183.
195. Roscher, R.; Bohn, B.; Duarte, M.F.; Garcke, J. Explainable Machine Learning for Scientific Insights and Discoveries. *IEEE Access* **2020**, *8*, 42200–42216, doi:10.1109/ACCESS.2020.2976199.
196. Voosen, P. The AI detectives. *Science* **2017**, *357*, 22–27, doi:10.1126/science.357.6346.22.
197. Roberts, M.G.; Lawrence, R. Chapter 3 MedChemInformatics: An Introduction to Machine Learning for Drug Discovery. In *Machine Learning in Chemistry: The Impact of Artificial Intelligence*; The Royal Society of Chemistry: London, UK, 2020; pp. 37–75. doi:10.1039/9781839160233-00037.

Chapter 2

A comprehensive methodology for the development of descriptor-based QSPR/QSAR models

Contents

2.1 Introduction

As discussed in Chapter 1, the choice of a ML method depends on the data set characteristics and problem requirements. To illustrate this, the following two chapters (Chapters 2 and 3) provide a first case study with the prediction of two thermodynamic properties of molecules based on their structural and physico-chemical features (or descriptors). This type of model is also known as Quantitative Structure-Property Relationship (QSPR) model (or Quantitative Structure-Activity Relationship (QSAR) when biological activities are predicted). The work is divided into two parts. Chapter 2 focuses on the development of the QSPR modeling procedure from data collection to model construction, with a multi-angle approach to provide an overview of the different options at each stage and to understand their effects on the final model. Chapter 3 addresses an important aspect in QSPR modeling which is the applicability domain definition, with methods adapted to high dimensional problems.

Why ML?
Firstly, the reason for using ML in this case study is that conventional methods for determining these properties (group contribution methods, quantum chemical methods) can only be applied to simple/small molecules, and require expert knowledge to apply them correctly. On the one hand, breaking down a molecule into sub-groups for which the thermodynamic properties are known becomes difficult for larger/more complex molecules. On the other hand, quantum chemistry methods are physically complex (as they are based on solving Schrödinger's equation) and time-consuming.

Constraints of the case study
In the context of this work, the development of the QSPR model is carried out under certain criteria/constraints. Indeed, the ultimate goal is to be able to apply the developed model to any molecule in the context of the discovery of new chemicals. Consequently, the database considered for training the model is made up of a very wide variety of molecular structures (23 families). Defining the model's applicability domain becomes therefore crucial, given that an ML model is not extrapolable. A second constraint is the lack of knowledge concerning the structural and physico-chemical characteristics relevant to predict the two thermodynamic properties. Thus, molecules are represented by a large number of descriptors (5666) to ensure the most complete representation and then, mathematical methods are implemented to identify the most relevant ones. Finally, the last criterion is based on OECD (Organisation for Economic Co-operation and Development) principles, which have been established to facilitate the consideration of QSPR/QSAR models in the regulation of chemical substances. These principles are: a defined endpoint, an unambiguous algorithm, a defined domain of applicability, appropriate measures of goodness-of-fit/robustness/predictivity and, if possible, a mechanistic interpretation.

Specificity of the work
The specificity of this work lies in the multi-angle approach adopted and the definition of the applicability domain within the framework of the criteria/constraints set out above. For example, many works are limited to a given chemical family of molecules. Others select descriptors based on their expertise of the system and/or a given method. Also, the choice and impact of different methods throughout the development of QSAR/QSPR models is not always obvious. Finally, the applicability domain is particularly difficult to determine in high-dimensional problems and adapted methods are lacking.

This chapter presents the first part of this case study, namely the multi-angle approach to

develop the QSPR model, from data collection to model construction. Different methods that are typically employed within the various steps of the QSPR modeling procedure are presented and evaluated. An overview of the investigated methods at the different steps is available in Figure 2.1.

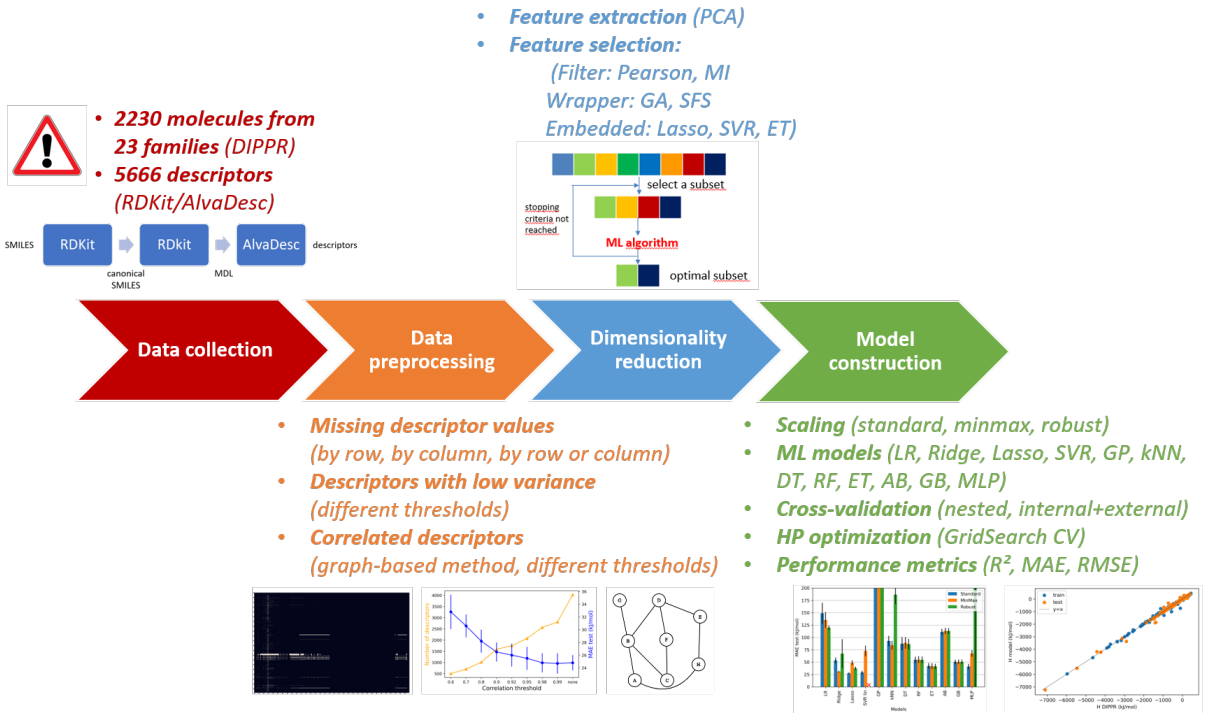


Figure 2.1: Overview of the methodology adopted in Chapter 2.

2.2 Outline of the publication

1. **Introduction**
2. **Data set and methods**
   - 2.1 Data set
   - 2.2 Descriptors
   - 2.3 Data preprocessing
   - 2.4 Dimensionality reduction
   - 2.5 ML model construction
3. **Results**
   - 3.1 Preliminary screening with default preprocessing and without dimensionality reduction
     - 3.1.1 Comparison of the performance of different models
     - 3.1.2 Comparison of data scaling and data splitting methods
   - 3.2 Effect of data preprocessing
   - 3.3 Effect of the dimensionality reduction
   - 3.4 Final ML modeling and HP optimization
4. **Benchmark**
5. **Conclusions and perspectives**

Abbreviations

References

2.3 Publication *"On the Development of Descriptor-Based Machine Learning Models for Thermodynamic Properties: Part 1 - From Data Collection to Model Construction: Understanding of the Methods and their Effects."*, published on 29 November 2023

processes

MDPI

Article

On the Development of Descriptor-Based Machine Learning Models for Thermodynamic Properties: Part 1—From Data Collection to Model Construction: Understanding of the Methods and Their Effects

Cindy Trinh, Youssef Tbatou, Silvia Lasala, Olivier Herbinet and Dimitrios Meimaroglou *

Université de Lorraine, CNRS, LRGP, F-54001 Nancy, France; cindy.trinh.ct@outlook.com (C.T.); yousseftbatou3@gmail.com (Y.T.); silvia.lasala@univ-lorraine.fr (S.L.); olivier.herbinet@univ-lorraine.fr (O.H.)
* Correspondence: dimitrios.meimaroglou@univ-lorraine.fr

Abstract: In the present work, a multi-angle approach is adopted to develop two ML-QSPR models for the prediction of the enthalpy of formation and the entropy of molecules, in their ideal gas state. The molecules were represented by high-dimensional vectors of structural and physico-chemical characteristics (i.e., descriptors). In this sense, an overview is provided of the possible methods that can be employed at each step of the ML-QSPR procedure (i.e., data preprocessing, dimensionality reduction and model construction) and an attempt is made to increase the understanding of the effects related to a given choice or method on the model performance, interpretability and applicability domain. At the same time, the well-known OECD principles for the validation of (Q)SAR models are also considered and addressed. The employed data set is a good representation of two common problems in ML-QSPR modeling, namely the high-dimensional descriptor-based representation and the high chemical diversity of the molecules. This diversity effectively impacts the subsequent applicability of the developed models to a new molecule. The data set complexity is addressed through customized data preprocessing techniques and genetic algorithms. The former improves the data quality while limiting the loss of information, while the latter allows for the automatic identification of the most important descriptors, in accordance with a physical interpretation. The best performances are obtained with Lasso linear models (*MAE* test = 25.2 kJ/mol for the enthalpy and 17.9 J/mol/K for the entropy). Finally, the overall developed procedure is also tested on various enthalpy and entropy related data sets from the literature to check its applicability to other problems and competing performances are obtained, highlighting that different methods and molecular representations can lead to good performances.

Keywords: machine learning; QSPR/QSAR; high-dimensional data; descriptors; thermodynamic properties; feature selection; genetic algorithms

Citation: Trinh, C.; Tbatou, Y.; Lasala, S.; Herbinet, O.; Meimaroglou, D. On the Development of Descriptor-Based Machine Learning Models for Thermodynamic Properties: Part 1—From Data Collection to Model Construction: Understanding of the Methods and Their Effects. *Processes* **2023**, *11*, 3325. https://doi.org/10.3390/pr11123325

Academic Editor: Antonino Recca

Received: 20 October 2023
Revised: 13 November 2023
Accepted: 17 November 2023
Published: 29 November 2023

1. Introduction

Quantitative Structure Property/Activity Relationship (QSPR/QSAR) models have been widely employed for several decades in chemistry-related fields to predict various endpoints of molecules (i.e., physico-chemical properties and biological activities, respectively) on the basis of their structure (e.g., descriptors, fingerprints, graphs), via mathematical methods. Successful QSPR/QSAR applications include very different endpoints such as critical temperature and pressure [1], normal boiling point [2], heat capacity [3], enthalpy of solvation [4]/vaporization [5,6], blood-brain barrier permeability [7], physico-chemical properties of polymers/fuels/ionic liquids [8–15], solubility [16–21], minimum ignition energy of combustible dusts [22] or antibacterial/antiviral properties [23,24].

To construct these QSPR/QSAR models, numerous mathematical methods were used ranging from simple and interpretable linear regression methods (e.g., multiple linear

regression and partial least squares) to more complex and nonlinear machine learning (ML) and deep learning methods, in response to the rising complexity of available data sets (e.g., larger data sets, nonlinear relations between molecular structures and endpoints, diversity in the molecular structures) [25–29]. Similarly, significant progress has been made in terms of the molecular structure representation, which evolved from simple representations (e.g., with few descriptors) to more complex ones (e.g., with up to thousands of descriptors, or based on graph neural networks (GNN)). More generally, the need to discover and develop more rapidly new molecules and properties has kept QSPR/QSAR research particularly active. These data-driven models effectively circumvent the complex and time-consuming development of knowledge-based models and experimental studies. More examples of artificial intelligence and ML application in various subfields of chemistry can be found in [30,31].

However, many QSPR/QSAR works lack important elements and fail to properly address the recommendations from the OECD (Organization for Economic Co-operation and Development) [25,32,33]. In particular, these recommendations are composed of 5 principles aiming at 'facilitating the consideration of a (Q)SAR model for regulatory purposes', for example when predicting the health hazards and toxicity of new chemicals [25,29]. These principles dictate that any relevant study should clearly include a defined endpoint, an unambiguous algorithm, a defined domain of applicability, appropriate measures of goodness-of-fit/robustness/predictivity and, if possible, a mechanistic interpretation [34]. Even if they were initially established to predict the hazards of chemicals, these general principles well-addressed the critical aspects during the development of any ML procedure. Besides, the use of ML methods has exploded over the last decades and there is a lack of "rules" to control whether the models are properly developed, which would facilitate their use and acceptance. Developing a ML model without possible further application is indeed useless. For all these reasons, the OECD principles were considered in this work in the case of thermodynamic properties.

The development of any ML-QSPR/QSAR model is generally composed of the following well-known steps: data collection, data preprocessing, dimensionality reduction, model construction and applicability domain definition. Along the implementation of these steps, a great number of methods and choices are presented to the developer, depending also on the characteristics of the problem and the available data, and these have a direct impact on the model performance, interpretability and applicability (e.g., to a new chemical). However, a clear overview of the possible methods or a clear justification of a choice over another one does not typically accompany relevant studies, thus making it unclear whether the proposed solution is general or robust enough for the envisioned application area. Accordingly, the first and main contribution of this work is to break-down and analyze the different steps of the development of a ML-QSPR/QSAR model in an attempt to assess the impact and the contribution of each choice and method along the process, while considering the OECD principles. The objective of this methodological approach will be the development of a predictive ML-QSPR model for two thermodynamic properties of molecules, namely the enthalpy of formation and the absolute entropy for the ideal gas state of molecules at 298.15 K and 1 bar. The representation of the molecules will be based on molecular descriptors.

The enthalpy of formation and entropy, which are the endpoints of interest in this study, are crucial to many chemical applications. In particular, they are required in the design of molecules, since they impact molecular stability; they are also present in the development of kinetic models and the prediction of reactions since they influence energy balances and equilibrium. Accordingly, the design of any process, involving chemical reactions or heat transfer, is prone to depend on the existence of accurate models for the prediction of these properties. Among the most common approaches to predict them, quantum chemistry (QC) and group contribution (GC) methods have been largely employed so far for their accuracy (e.g., $<$1 kcal/mol for the enthalpy of formation of small molecules) and/or simplicity [35–44]. However, for large/complex molecules, QC methods become

physically and computationally complex, while for GC methods, the decomposition of the molecules into known groups becomes a tedious/infeasible task and corrections due to the contribution of the 3D overall structure are needed (e.g., to include steric effects and ring strain effects). Consequently, ML methods represent an interesting alternative to the aforementioned QC and GC approaches due to their accuracy, low computation time and ability to describe complex problems without requiring physical knowledge. At the same time, ML methods, being data-driven in nature, suffer from a lack of interpretability and extrapolability, in comparison to their QC and GC knowledge-based counterparts [45].

Molecular descriptors represent diverse structural and physico-chemical characteristics of the molecules. Thousands of different descriptors have been reported in the literature and their calculation is nowadays facilitated by the use of publicly accessible libraries and software (e.g., RDKit, AlvaDesc, PaDEL, CDK, Mordred) [46–50]. In particular, the software AlvaDesc, which was employed in the present study, generates a total of 5666 descriptors for each molecule. This relatively high number of descriptors (i.e., concerning a physico-chemical problem) contains rich information on the molecular structures and thus increases the chances of capturing the relevant features affecting the thermodynamic properties, in the absence of knowledge. At the same time, this poses a number of difficulties in the development of the ML-QSPR/QSAR model and its generalized implementation and interpretation. These difficulties are related to the need to distinguish, at a certain point within the development procedure, the number and identity of the most relevant descriptors to the endpoints of interest, which remains one of the biggest challenges related to the use of descriptors (a.k.a. the "curse of dimensionality"). Commonly, to overcome these issues, a dimensionality reduction step is implemented before the model construction. On the one hand, feature extraction methods project the original high-dimensional space into a new space of lower dimension, thus creating new features being linear or nonlinear combinations of the original ones. On the other hand, feature selection methods select only a limited subset of descriptors as being the most representative ones and the rest are discarded, which facilitates the interpretability of the subsequent model in comparison with feature extraction methods. The selection of descriptors can also be based on available knowledge (i.e., expert input) but such knowledge is not readily available for the complete list of generated descriptors. These difficulties and the different dimensionality reduction approaches that can be undertaken under the premise that physical knowledge is not available a priori for all 5666 descriptors are analyzed as part of this work. A mechanistic interpretation of the descriptors that are identified as highly relevant by the different approaches is also attempted.

Finally, this study was not constrained to molecules belonging to a limited number of chemical families and structures, but, within the perspective of the discovery of new molecules for various applications, the development of models that will be applicable to a large diversity of molecules was pursued. Note, that this is a specific differentiating point of the present study and a major challenge as many reported studies are restricted to molecules of specific chemical families and/or structural characteristics [51–57].

More generally, this work constitutes a multi-angle, holistic approach to the procedure for the development of generally-applicable ML-QSPR/QSAR models, based on a high-dimensional representation of molecules (i.e., descriptors) and in the presence of limited expert-domain knowledge, following the recommendations of the OECD. As such, it can serve to enlighten different aspects of the process, especially the ones that are poorly discussed in the literature, as well as to guide newcomers in the field. To facilitate legibility, the presentation of the complete study will be made through a series of articles, the present one being the first of the series and focusing on the general methodology from data collection to model construction. The following article addresses the questions of defining the applicability domain and detecting the outliers at different stages of the ML-QSPR procedure, this challenge being related to the high-dimensional molecular representation.

2. Data Set and Methods

This section provides detailed information about the employed data set and methods, in agreement with the first and the second principles of the OECD, namely "a defined endpoint" and "an unambiguous algorithm".

2.1. Data Set

DIPPR's (Design Institute for Physical Properties) Project 801 Database (version 05/2020) [58], containing 2230 molecules represented by their Simplified Molecular Input Line Entry Specification (SMILES), was employed in this work. A large diversity of molecules is included in this database in terms of chemical family, size, atomic composition and geometry (e.g., linear/cyclic/branched, simple/multiple bonds). Figure 1 presents the distribution of the molecules of the database in terms of their chemical family. It can be observed that the number of molecules varies significantly among the different chemical families (i.e., between 15 molecules for inorganic compounds and 247 molecules for halogen compounds). Figure 2a shows the respective atom number-distribution of the same molecules. The vast majority of molecules, corresponding to ca. 90% of the database, have less than 40 atoms while the number of large molecules (i.e., containing more than 100 atoms) is limited to 15 molecules. Figure 2b provides additional information on the number of cycles of the molecules of the database. It can be observed that highly-cyclic molecules are under-represented in this specific database. Additional ways to compare the molecules could be envisioned but the presented figures suffice to demonstrate the high degree of heterogeneity that characterizes the data set. Note that the following molecules were eliminated due to identical SMILES and, hence, identical descriptors: hydrogen/hydrogen (para), phosphorus (white)/phosphorus (red) and cis-1,8-terpin/trans-1,8-terpin. Deuterium, perchloryl fluoride, chlorine trifluoride and air were eliminated as well from the database due to technical issues in calculating their descriptor values. The resulting dataset is then composed of 2220 molecules.

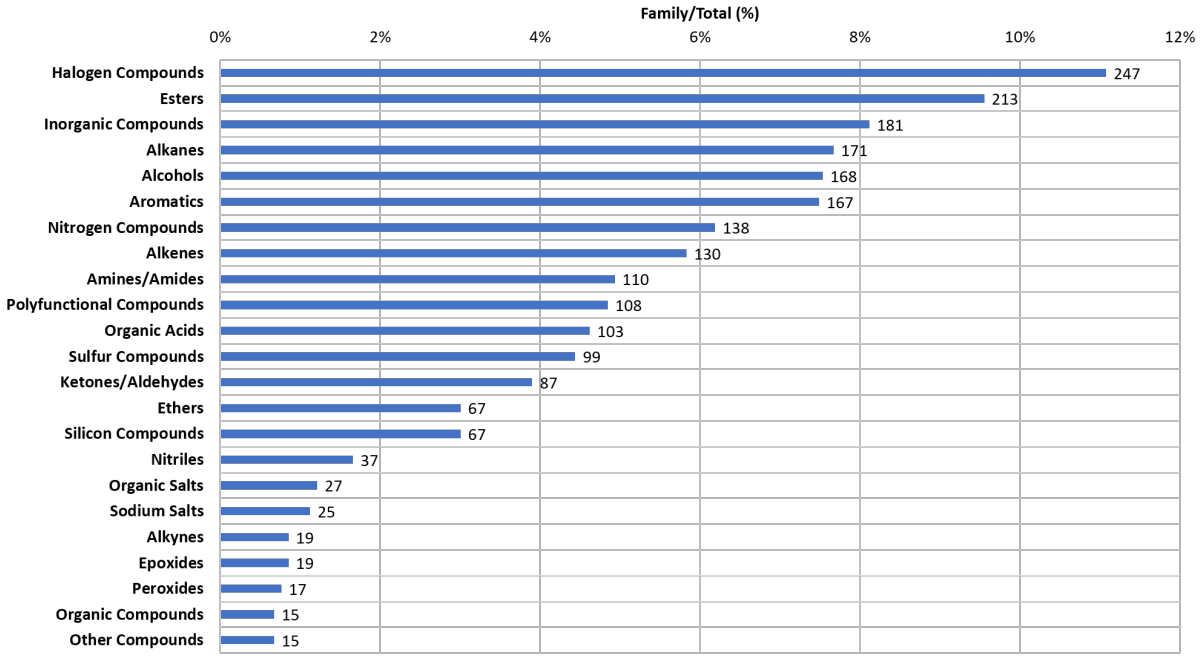
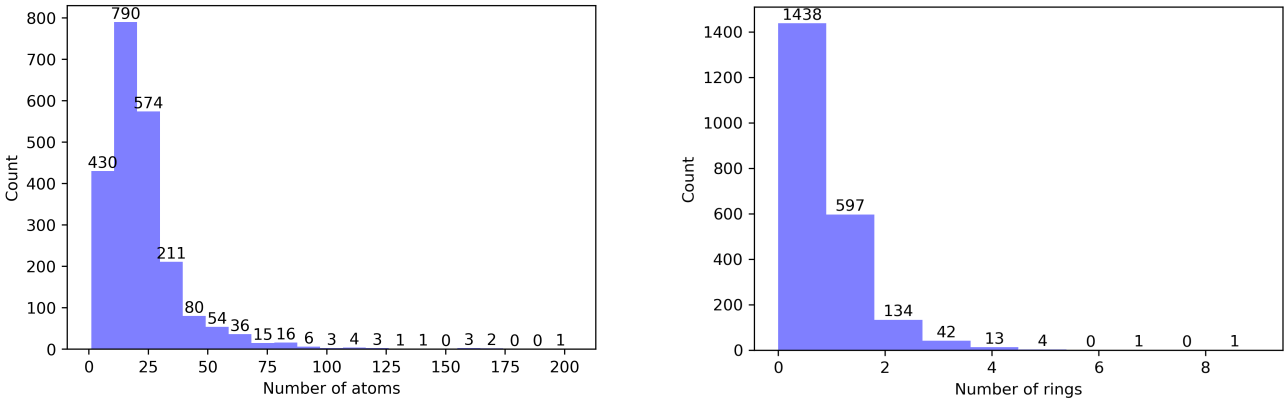


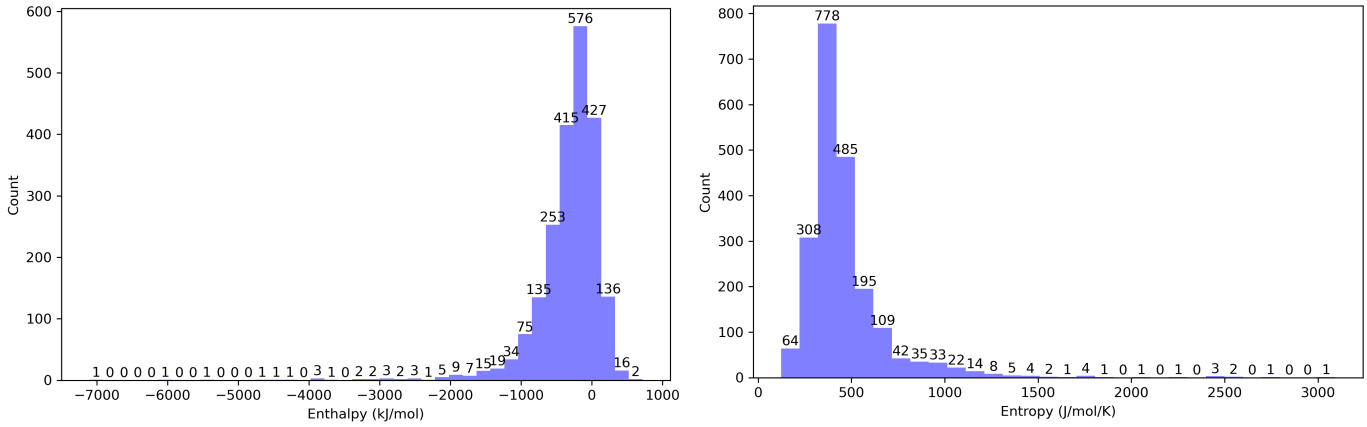
Figure 1. Classification of DIPPR molecules per chemical family (the numbers on the right of the bars correspond to the number of molecules within each family).



(**a**) (**b**)

Figure 2. Classification of DIPPR molecules (**a**) per number of atoms and (**b**) per number of rings, within each molecule.

In this work, the considered endpoints are the enthalpy of formation and the absolute entropy for the ideal gas state of molecules at 298.15 K and 1 bar. For simplicity, they will be henceforth, respectively, denoted as enthalpy (H) and entropy (S). For each molecule of the database, the values of these physico-chemical properties are accompanied by the associated determination method and the relative uncertainty. Diverse determination methods have been used in the construction of the database including both theoretical calculations (e.g., QC, GC, calculations based on other phases, conditions or properties) and experimental measurements. The distribution of the values of both properties is given in Figure 3. The relative uncertainties are classified in different levels within the DIPPR database, namely <0.2%, <1%, <3%, <5%, <10%, <25%, <50%, <100% and NaN, as shown in Table 1. This classification depends on several criteria such as data type, availability, agreement of data sources, acquisition method or originally reported uncertainty [59]. In this work, only the molecules within the five first classes of relative uncertainties were considered as a compromise between the number of molecules and data reliability. Accordingly, the resulting data sets for the enthalpy and entropy were composed of 1903 and 1872 molecules, respectively.



(**a**) (**b**)

Figure 3. Distribution of (**a**) the enthalpy and (**b**) the entropy values of the DIPPR database. A total of 2147 and 2119 values are present in the database for the enthalpy and the entropy, respectively.

Table 1. Classification of DIPPR data per uncertainty.

Property	Uncertainty								
	<0.2%	**<1%**	**<3%**	**<5%**	**<10%**	**<25%**	**<50%**	**<100%**	**NaN**
Enthalpy	50	401	1013	242	197	188	33	4	19
Entropy	66	184	1019	419	184	199	20	0	28

2.2. Descriptors

There are different ways to represent molecular structures such as SMILES, fingerprints, descriptors or graphs [60]. Each representation has its own advantages and drawbacks and the choice will depend on each problem's requirements and characteristics. In particular, the use of graph-based representations has exploded over the last decade due to their ability to learn the relevant chemical features, thus preventing the manual feature engineering step of traditional representations (e.g., descriptors or fingerprints) [61,62]. Nevertheless, this works focuses on descriptor-based representations for their simplicity and easier interpretability, while displaying good performances in various works [63–65]. There is no consensus about the best molecular representation yet (i.e., leading to the best prediction accuracy), and different representations can lead to comparable predictions [63,64,66]. Indeed, each representation contains different information about the molecular structure and it is difficult to know which information is relevant for a given property. In any case, the comparison and/or combination of descriptors with other molecular representations can be envisioned as a future step of this work.

Molecular descriptors consist of different numerical properties, characteristic of the structural and topological features or other physico-chemical properties of the molecules, that are commonly employed in similar QSPR/QSAR studies. In this study, descriptors were used instead of SMILES to represent the molecules as they contain 2D (based on molecule graph) and 3D (based on 3D coordinates) information which could impact the properties of interest. Indeed, enthalpy and entropy, respectively, measure the heat content and disorder of a molecule and are, therefore, sensitive to its structure.

The values of the descriptors can be calculated by means of different libraries or software, such as PaDEL [48], RDKit [46], CDK [49], AlvaDesc [7,47] or Mordred [50], on the basis of a standardized description of the molecules (i.e., as input), such as their SMILES notation. In this work, two open-source (PaDEL and RDKit) and one closed-source (AlvaDesc from Alvascience) tools were tested. Among them, AlvaDesc was finally retained, mainly due to the high number of calculated molecular descriptors it provides (i.e., 5666 descriptors were provided by AlvaDesc), as well as due to its robustness, ease of implementation, execution speed and proposed documentation and support. A comparison of different relevant libraries and software can be found in [50]. Note, that in AlvaDesc software, 1500 3D descriptors require information that can not be provided via the SMILES notation (i.e., related to the 3D atoms coordinates of the molecules). It was, therefore, necessary to convert the SMILES notation of the molecules to an MDL Mol standard, prior to importing them into AlvaDesc. The MDL Mol format essentially consists in an atom block which describes the 3D coordinates of each atom of the molecule, and a bond block which indicates the type of bonds between the atoms. The whole conversion procedure is summarized in Figure 4. The conversion of SMILES (from DIPPR) to MDL Mol format was performed in two steps, using RDKit, an open-source toolkit for cheminformatics; first, the SMILES notation from DIPPR was converted to canonical SMILES, the latter being unique to each molecule as opposed to SMILES. Then, in order to convert canonical SMILES to the MDL Mol format, the RDKit was employed and generated the conformers of the molecules by applying distance geometry calculations. The conformers are subsequently corrected by the ETKDG (Experimental-Torsion Distance Geometry with additional basic knowledge terms) method of Riniker and Landrum, based on torsion angle preferences [67]. The ETKDG method, which is a stochastic method using knowledge-based and distance geometry algorithms, is considered to be an accurate fast conformer generation method, especially

for small molecules [68]. Lastly, once the MDL Mol format was generated, AlvaDesc was employed to calculate the 5666 descriptors for each molecule.

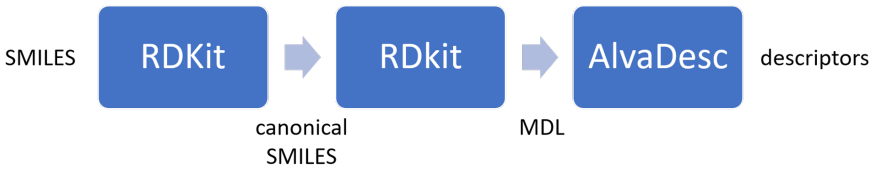


Figure 4. Procedure for converting the initial SMILES notation, of the DIPPR database, to molecular descriptor values.

The generated descriptors can be classified into 33 categories, as shown in Table 2. Their calculation is based on different mathematical algorithms, available in the literature. Some of them were developed on the basis of small organic molecules but the algorithms used in AlvaDesc software are considered to be applicable to a larger set of molecules [47]. Prior to the calculation of descriptors, AlvaDesc operates a series of internal standardization procedures on molecular structures to handle nitro groups, aromatization and implicit hydrogens. Other standardization procedures can be implemented via other tools (e.g., AlvaScience software, researcher knowledge) but are not in the scope of this work. However, this standardization step and, more generally, the accuracy in the representation of the molecular structures can highly impact the performance of the developed models, hence specific studies are reported on the preparation of chemical data [25,28,69].

Table 2. AlvaDesc descriptors per category.

Category n°	Category Name	Number of Descriptors	Category n°	Category Name	Number of Descriptors
1	Constitutional indices	50	18	WHIM descriptors	114
2	Ring descriptors	35	19	GETAWAY descriptors	273
3	Topological indices	79	20	Randic molecular profiles	41
4	Walk and path counts	46	21	Functional group counts	154
5	Connectivity indices	37	22	Atom-centred fragments	115
6	Information indices	51	23	Atom-type E-state indices	346
7	2D matrix-based descriptors	608	24	Pharmacophore descriptors	165
8	2D autocorrelations	213	25	2D Atom Pairs	1596
9	Burden eigenvalues	96	26	3D Atom Pairs	36
10	P_VSA-like descriptors	69	27	Charge descriptors	15
11	ETA indices	40	28	Molecular properties	27
12	Edge adjacency indices	324	29	Drug-like indices	30
13	Geometrical descriptors	38	30	CATS 3D descriptors	300
14	3D matrix-based descriptors	132	31	WHALES descriptors	33
15	3D autocorrelations	80	32	MDE descriptors	19
16	RDF descriptors	210	33	Chirality descriptors	70
17	3D-MoRSE descriptors	224			

2.3. Data Preprocessing

Data preprocessing is a step that, although time-consuming, is crucial in any ML-development project since the accuracy, efficiency and robustness of the developed model depend directly on the existence of sufficient data of high quality (i.e., without missing, constant, redundant, irrelevant values), as commonly transcribed by the popular concept of "garbage in, garbage out".

A preliminary analysis of the available data set revealed the following issues: (i) missing descriptor values (cf. Figure 5), (ii) descriptors with low variance (i.e., quasi-constant values for all molecules), and (iii) significant correlation between descriptor values (N.B. if two descriptors are highly correlated, only one of them could be sufficient to describe the property of interest, the other being redundant). The order in which these issues

will be dealt with, during the preprocessing stage, as well as the selected treatment approach for each issue, can influence the final (i.e., preprocessed) data set, and therefore, the performance of the model. In the present work, the following order was employed:

1. Elimination of missing descriptor values (Desc-MVs).
2. Elimination of descriptors with low variance.
3. Elimination of correlations between descriptors.

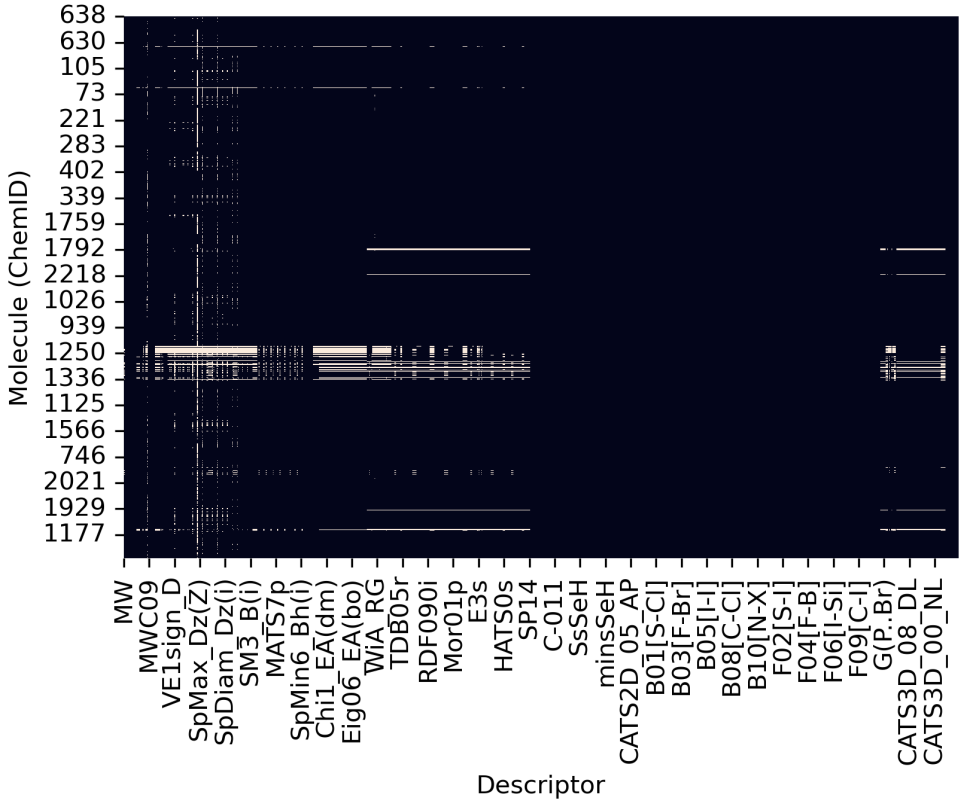


Figure 5. Heatmap of DescMVs (white = Desc-MVs; black = defined values; molecules are classified by their chemical family).

The elimination of the Desc-MVs was selected to be performed at the beginning of the preprocessing stage to ensure the unbiased calculation of the variance and of the correlation coefficients of the descriptor values, which are necessary for the subsequent steps. The existence of Desc-MVs in the data set is the result of the incapacity of AlvaDesc to calculate them for certain molecules, due to constraints related to the respective calculation algorithms (e.g., disconnected structures). Accordingly, their removal was preferred over the implementation of data-imputation techniques (e.g., mean, median, interpolation, ML), as the latter would risk introducing bias and artifact values into the data set. The three following algorithms were compared for the elimination of Desc-MVs: (i) elimination by rows (i.e., molecules), (ii) elimination by columns (i.e., descriptors), and (iii) alternating elimination by row or column.

The first algorithm removes all molecules that contain at least one missing value, presenting the drawback of a vast reduction of the number of considered molecules. The second algorithm consists of eliminating the complete descriptor from the data set, for all molecules, if this descriptor contains even a single missing value for a given molecule. This approach presents the inconvenience of eventually reducing the number of descriptors to a stage where molecules become identical among them, due to the loss of the differentiating descriptors. Note, that the important diversity of the considered molecules results in an inevitable absence of some values for certain groups of descriptors and for specific chemical families (cf. Figure 5), which is one of the challenging elements of the adopted generic (i.e.,

non family-specific) approach. The third algorithm was based on an iterative alternating step-wise elimination of either the molecule or the descriptor that contained the highest number of missing values at the given iteration, thus limiting the loss of information, both in terms of molecules and descriptors. In this latter elimination algorithm, iterations are carried on until the removal of all the Desc-MVs from the data set.

Concerning the elimination of descriptors with low variance, this was performed before the elimination of the correlations to reduce the computational cost associated with the calculation of the correlation matrix, required for the correlation elimination step. More generally, the role of this step is to remove the quasi-constant descriptors as they show no effect on the target property. Several threshold values were tested in terms of the minimum descriptor variance, below which the descriptor elimination should be employed. These threshold values are 0 and 10^k (for k in $\{-4, -3, -2, -1, 0, 1, 2, 3\}$).

Finally, the elimination of correlations between descriptors was based on the calculation of the correlation matrix among all descriptors. A novel approach, based on the graph theory, was employed to ensure that all correlations above a fixed threshold would be efficiently removed, without any additional information loss and without the risk of retaining redundant information in the data set. This approach is particularly pertinent in the presence of high-dimensional data sets, for which a pairwise consideration would be insufficient. Indeed, in an approach where correlated descriptors would be removed in consecutive loops of pairwise eliminations, one risks eliminating excessive information or even adding bias to the data set (cf. Supplementary Materials). According to the approach adopted here, it is possible to construct graphs in which nodes and edges represent descriptors and correlation coefficients, respectively. The designed procedure consists of selecting which descriptors to keep/remove in each graph, in order to eliminate all correlations above a fixed threshold value of the correlation coefficient, without losing additional information. Accordingly, the three following cases are distinguished, as also illustrated in Figure 6:

1. A descriptor does not belong to any graph (i.e., it is not correlated to any other descriptor) and must be retained.
2. Two descriptors form a complete graph. In this case, only one of them is retained.
3. Three or more descriptors belong to a graph. In this case, the descriptor with the most correlations is retained and all descriptors connected directly (i.e., descriptors that are nodes on common edges with the descriptor in question) with this one are eliminated. The remaining descriptors are analyzed through cases 1, 2 and 3 until there is no descriptor left.

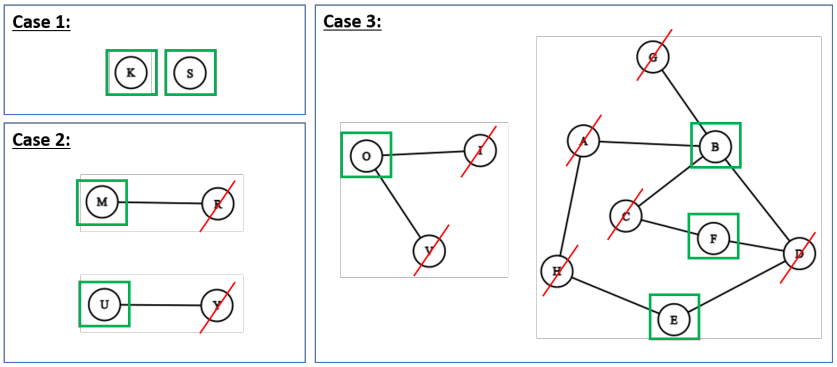


Figure 6. Graph theory-based method for the elimination of correlations between descriptors (nodes and edges correspond to descriptors and correlations (above a given threshold for the value of the correlation coefficient), respectively). **Case 1**: non correlated descriptors; **Case 2**: pairwise correlated descriptors; **Case 3**: multiple correlations between descriptors. Descriptors in green are selected while those in red are removed.

The following thresholds were tested to eliminate correlations between descriptors: 0.6, 0.7, 0.8, 0.9, 0.92, 0.95, 0.98 and 0.99.

All the configurations that were tested during the preprocessing step, in the framework of the present study, are summarized in Table 3. Default values for the different preprocessing steps were also set up for a preliminary screening of various ML methods in Section 3.1.

Table 3. Summary of the tested and default preprocessing options.

Preprocessing Step	Tested	Default
Elimination of Desc-MVs	- By row - By column - Alternating row or column	Alternating row or column
Elimination of descriptors with low variance	[0, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]	0.001
Elimination of correlated descriptors	[0.6, 0.7, 0.8, 0.9, 0.92, 0.95, 0.98, 0.99]	0.95

2.4. Dimensionality Reduction

Prior to applying directly ML models to the preprocessed data, it can be necessary to reduce the number of descriptors through dimensionality reduction methods. Indeed, this step helps to reduce the computational cost associated with the model implementation, prevent overfitting and eventually improve interpretability by identifying the most relevant descriptors. It also allows to increase the ratio of training molecules to descriptors which further strengthens the model significance and reduces variance [25,70]. Dimensionality reduction methods can be divided into two categories, namely feature extraction and feature selection methods. The former creates a lower dimensional set of new descriptors, consisting of combinations of the original descriptors. Principal component analysis (PCA), linear discriminant analysis (LDA) and autoencoders are examples of popular feature extraction methods [71–75]. Conversely, feature selection methods are based on the premise of selecting a subset of the original descriptors, without transforming them, and are typically distinguished into three sub-categories, as illustrated in Figure 7: filter, wrapper and embedded methods [76–80].

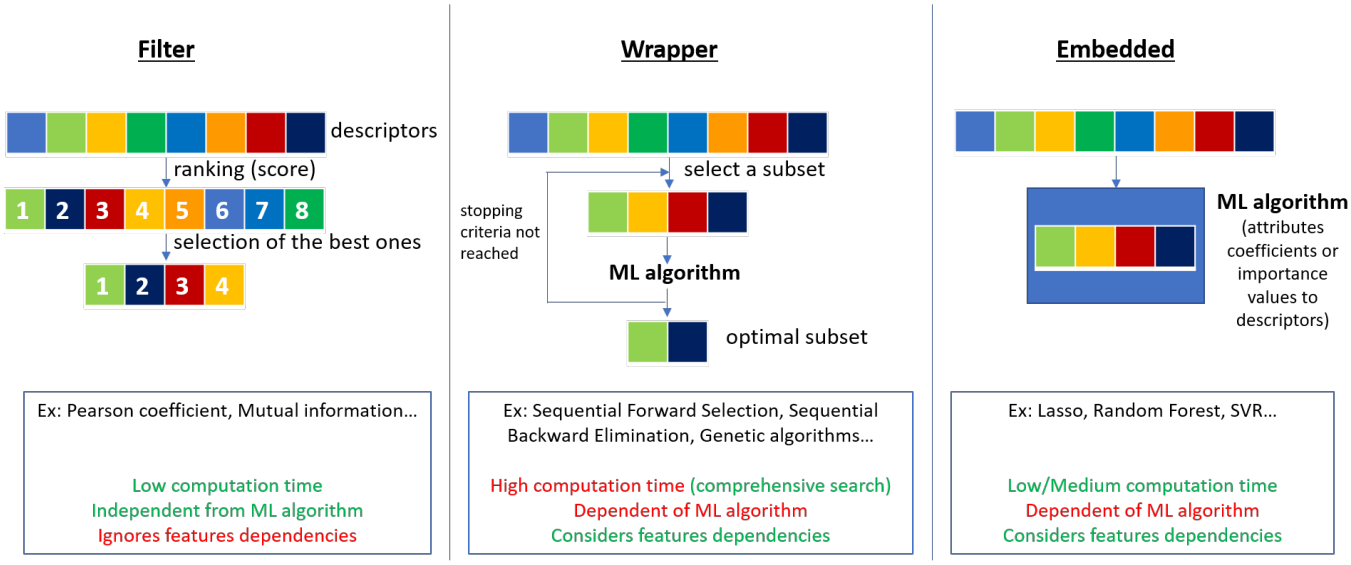


Figure 7. Overview of feature selection methods with their advantages and limits in green and red, respectively.

Filter methods calculate a score for each descriptor, without implementing a ML model, and use these scores to rank descriptors and select those whose values are situated above/below a given threshold. The calculation of the Pearson coefficient between each descriptor and the response is a typical example of such an approach. To some extent, the elimination of descriptors with low variance, as presented previously within the data preprocessing step, can also be considered as a filter method since the values of the variance served to 'filter' the descriptors. Inversely, wrapper and embedded methods both require (and hence, depend on) the implementation of a ML model. Concerning wrapper methods, they consist of evaluating different possible subsets of descriptors (through the selected ML model) until a stopping criterion is fulfilled (e.g., related to the number of descriptors or to the performance of the ML model). Genetic algorithms (GA) and sequential approaches (e.g., sequential forward selection (SFS) or backward elimination) belong to wrapper methods. As for embedded methods, their name originates from the fact they are 'embedded' in the selected ML model, meaning that the latter internally identifies the most important descriptors during the training phase. The importance of each descriptor can be read through some attributes of the ML model such as the weights/coefficients in regression models (e.g., least absolute shrinkage and selection operator (Lasso), support vector regression (SVR)) or the impurity-based feature importance in ensemble models (e.g., random forest (RF), extra trees (ET)).

More generally, feature selection methods find wider application in QSPR/QSAR studies than feature extraction ones, in which high-dimensional data sets are more often encountered, particularly in bioinformatics for the selection of genes [76,81,82]. Indeed, in high-dimensional problems, it is particularly difficult to interpret extracted features that are expressed in the form of combinations of an important number of descriptors, as part of a feature extraction approach. Within feature selection methods, wrapper approaches are more likely to find a suitable subset of descriptors, respecting the imposed criteria, as a result of a comprehensive search in the descriptor space. Additionally, although wrapper and embedded methods depend on the choice of a specific ML model, they consider dependencies between descriptors which can help to improve ML model performance in comparison to filter methods. However, both come at the expense of a higher computation time, although embedded methods generally offer a better compromise between computation time and ML model performance. Note that, a great diversity of feature selection methods/categories is reported in the literature, extending well beyond the brief overview attempted in this work, while their implementation may also include sequential combinations of different techniques [76,77,81,83–87].

As part of this study, different dimensionality reduction methods were tested and compared, as summarized in Table 4: PCA for feature extraction as well as two filter methods (Pearson coefficient and mutual information (MI)), two wrapper methods (GA and SFS) based on Lasso model and three embedded methods (Lasso, SVR lin and ET) for feature selection. All these methods were implemented using the Scikit-learn default options of Python v3.9.12 [88], except for the GA whose procedure is fully described in the Supplementary Materials. In all cases, a mechanistic interpretation of the identified descriptors with the different dimensionality reduction methods is attempted, in compliance with the scope of this study and the fifth principle of the OECD.

Table 4. Methods tested for dimensionality reduction.

Feature Extraction	**Feature Selection**		
	Filter	**Wrapper**	**Embedded**
PCA	Pearson coefficient MI	GA SFS	Lasso SVR lin ET

2.5. ML Model Construction

This step, which is the central one in the study, consists of training and subsequently validating ML models on the basis of the final form (i.e., after the preprocessing and dimensionality reduction steps) of the data set. Once again, the developer is faced with a series of dilemmas, both in terms of the selection of the most appropriate ML methods as well as in terms of their implementation options, such as the ones concerning data scaling, data splitting, optimization of the hyperparameters (HPs), selection of the most suited performance metrics, etc. All the configurations that were tested during this step, in the framework of the present study, are summarized in Table 5.

Table 5. Configurations tested during ML model construction.

Data Scaling	Data Splitting	ML Models	Performance Metrics
Standard	5-fold internal CV	LR	Coefficient of determination
$x_{scaled} = \frac{x-\bar{x}}{\sigma_x}$	5 and 10-fold external CV	Ridge	$R^2 = 1 - \frac{\sum(y_{DIPPR}-y_{predicted})^2}{\sum(y_{DIPPR}-\overline{y_{DIPPR}})^2}$
		Lasso	
Min-Max		SVR lin	Root Mean Squared Error
$x_{scaled} = \frac{x-x_{min}}{x_{max}-x_{min}}$		GP	$RMSE = \sqrt{\frac{1}{n}\sum(y_{DIPPR} - y_{predicted})^2}$
		kNN	
Robust		DT	Mean Absolute Error
$x_{scaled} = \frac{x-Q_2}{Q_3-Q_1}$		RF	$MAE = \frac{1}{n}\sum\left\|y_{DIPPR} - y_{predicted}\right\|$
		ET	
		AB	
		GB	
		MLP	

Data scaling consists of transposing the values of all the input features (i.e., the descriptors in this case) to a reference range before training, so that their original differences in scale are not considered by the model as significant. Although this scaling step is considered a rather trivial procedure in all data-driven modeling studies, depending on the type of ML method, it may affect the performance of the model. In this work, different scaling methods were compared, namely the standard, min-max and robust scaling techniques (cf. Table 5). The latter is characterized by its robustness to outliers, as it is based on quartiles, while the two former are more sensitive to outliers since their calculation is based on the mean, standard deviation, min and max values. Note, that an outlier is loosely referred here to an abnormal observation among a set of values (e.g., descriptor values, response values). A more detailed discussion on the identification and treatment of outliers is included in the second article of this study [89].

Data splitting is the partitioning of data into training, validation and test sets. In particular, a nested cross-validation (CV) scheme was employed in this work to assess the effect of data splitting on the model performance (represented by error bars or uncertainties in the graphs and tables of this article), and therefore, produce more significant and unbiased performance estimates [32,70,90,91]. As shown in Figure 8, the nested CV procedure is effectively composed of an internal k-fold CV loop, nested within an external k′-fold one. The former is used for the optimization of the HPs while the latter is for model selection. Concerning the selection of the values of k and k′, these depend on the quantity of data and affect the simulation time since a higher value of k (or k′) will require a higher number of simulation passes. The most commonly encountered values are 5 or 10 as they have been found to ensure a good trade-off between the amount of training data, bias, variance and computation time [92,93]. In this work, k was fixed at a value of 5 while the value of k′ was varied between 5 and 10 to assess its impact on the performance of the developed models.

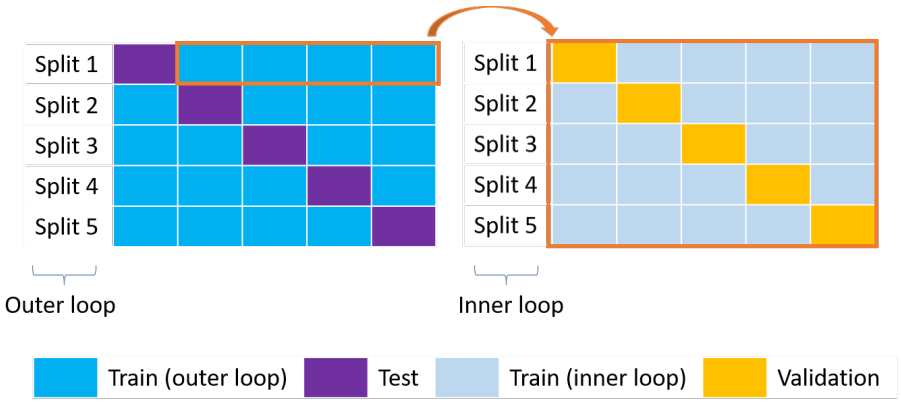


Figure 8. Nested CV. The outer loop on the left (blue and purple boxes for the training and test sets, respectively) is used for model selection while the inner loop on the right (grey and yellow boxes for the training and validation sets, respectively) is used for the optimization of the HPs.

Note, that in an attempt to minimize data leakage in this work, only the training data set (from the external loop) was used to determine the parameters of the scaling methods but also during the earlier dimensionality reduction step. The term "data leakage" describes cases in which model training uses, implicitly or explicitly, information that is not strictly contained in the training data set. For example, during a standard scaling of the data, if the mean and the standard deviation are calculated on the complete data set (i.e., including the test data), this information about the test data is implicitly included in the model training process. If not well addressed, and depending on the data distribution, data leakage can lead to highly-performing models on the data set but with limited generalization capacities.

Accordingly, the effects of the different scaling and splitting methods were evaluated for 12 linear and nonlinear ML models. These include ordinary least squares linear regression (LR), ridge, Lasso, SVR lin (SVR lin), Gaussian processes (GP), k-nearest neighbors (kNN), decision tree (DT), RF, ET, gradient boosting (GB), adaptive boosting (AB) and multilayer perceptron (MLP). Among the most popular performance metrics, which are typically employed to evaluate and compare models, are the coefficient of determination, R^2, the root mean squared error, $RMSE$, and the mean absolute error, MAE (cf. Table 5). Other examples of metrics that are employed in similar studies can be found in [32]. The choice of the most pertinent performance metric that will help discriminate models depends on the problem requirements; for example, if high prediction errors must be penalized at all costs (i.e., even for acceptable overall average performances), $RMSE$ will be more adapted than MAE. In this article, the three aforementioned performance metrics will be provided separately for the internal training and validation and the external training and test sets, to facilitate comparison with other similar studies. The computation times will also be provided, as they can constitute an additional decision criterion.

More generally, the evaluation of the performance of a model is to be related to the fourth principle of the OECD, concerning the implementation of "appropriate measures of goodness-of-fit/robustness/predictivity". The two former refer to the model internal performance, in terms of the training set, while the latter refers to the external performance, in terms of the test set. In particular, the goodness-of-fit measures how well the model fits with the data, the robustness is the stability of the model in case of a perturbation (e.g., modification of the training set via CV methods) and the predictivity measures how accurate the prediction for a new molecule is [34]. Many statistical validation techniques other than the CV method used in this work can be found in the literature [28,34]. Besides, the identification of appropriate metrics for external validation has been much debated; for example, the suitability of R^2 as an appropriate metric for such studies has been criticized as it only measures how well the model fits the test data [28,94,95]. In general, the use of several metrics is recommended and a model can be accepted if it performs well in

all metrics (i.e., displaying high R^2 and low MAE and $RMSE$ values) for all training, validation and test sets.

The performance of ML models can be further improved via an optimization step of their HP values. These are parameters that define structural elements of the methods, such as the number of neurons or hidden layers in MLP, and whose values are not determined as part of the training phase. In this respect, GridSearch CV was employed in this work to optimize the HPs of the ML models that were identified as best-performing ones, after an initial screening stage. This technique consists of evaluating the different possible combinations of HP values, given a grid of predefined ranges for each one by the user. Other methods, sometimes more adapted to specific ML models, are also reported in the literature [96] but their exhaustive evaluation was found to exceed the scope of this work.

All the ML models of this work were implemented using the Scikit-learn library v1.0.2 of Python v3.9.12 [88], while RDKit v2022.03.5 and AlvaDesc v2.0.8 were used for the generation of the data set. All the reported simulation times concern runs that were carried out on an Intel® Core™i9-10900 CPU @2.80 GHz personal workstation.

3. Results

For reasons of brevity, all the figures and tables of results that are provided in this section concern the modeling of the enthalpy, unless otherwise indicated. Those for the entropy are provided in the Supplementary Materials.

3.1. Preliminary Screening with Default Preprocessing and without Dimensionality Reduction

3.1.1. Comparison of the Performance of Different Models

Before investigating the effects of data preprocessing and dimensionality reduction, a preliminary screening of different ML modeling methods is performed to quickly identify the most promising ones for the present regression problem. This will allow also to evaluate the effects of data scaling and splitting methods, as well as to assess the pertinence of the selected performance metrics. The performances (R^2, MAE and $RMSE$) of the 12 screened ML models are given in Figure 9 for the external training and test sets, the error bars corresponding to different splits. These values are obtained with data containing 1785 molecules and 1961 descriptors, resulting from the previously described preprocessing steps with the default options (cf. Table 3). Furthermore, the steps of dimensionality reduction and HP optimization are omitted in this preliminary screening. All data are scaled with the standard method and split according to a 5-fold external CV (i.e., approx 1428 (80%) molecules for training and 357 (20%) for testing).

Based on the different performance metrics, the models displaying the best generalization (i.e., test) performances are Lasso, SVR lin, ET and MLP. Their parity plots are displayed in Figure 10. Figure 9 shows that the linear regression models Ridge and Lasso both perform better than LR, all three models being defined by the general Equation (1):

$$\widehat{y} = w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_px_p + b = Xw + b \quad (1)$$

where $\widehat{y}$ is the vector of predicted values, $w = (w_1 \ldots w_p)$ corresponds to the parameters (a.k.a. coefficients or weights) of the model, $X = (x_1 \ldots x_p)$ is the design matrix of size (n, p) with n and p the number of molecules and descriptors, respectively, and b is the intercept.

The superior performance of Ridge and Lasso, compared to LR, can be explained by the fact that their objective functions (cf. Equations (3) and (4), respectively) contain a regularization term, α, as opposed to that of LR (cf. Equation (2)). This regularization term penalizes the weights/coefficients of the input terms, X (i.e., corresponding to the descriptors), that do not display a significant contribution to the predicted property. The penalization takes the form of a value reduction that may result in complete elimination (i.e., shrinkage to zero) of some coefficients. This allows keeping the model as simple as possible and, hence, avoiding overfitting. At the same time, it can be shown that the L1-regularization, employed in Lasso, results in a higher elimination rate than the

L2-regularization, employed in Ridge [97]. Indeed, in the simulation shown here, Lasso eliminated around 88% of the 1961 descriptors while Ridge eliminated less than 1%. The adjustment of the value of the regularization coefficient, α, which is a HP of these models, determines the compromise between underfitting (i.e., the model is oversimplified) and overfitting (i.e., the model remains highly complex). Note that the Scikit-learn default value of $\alpha = 1$ was used in these simulations.

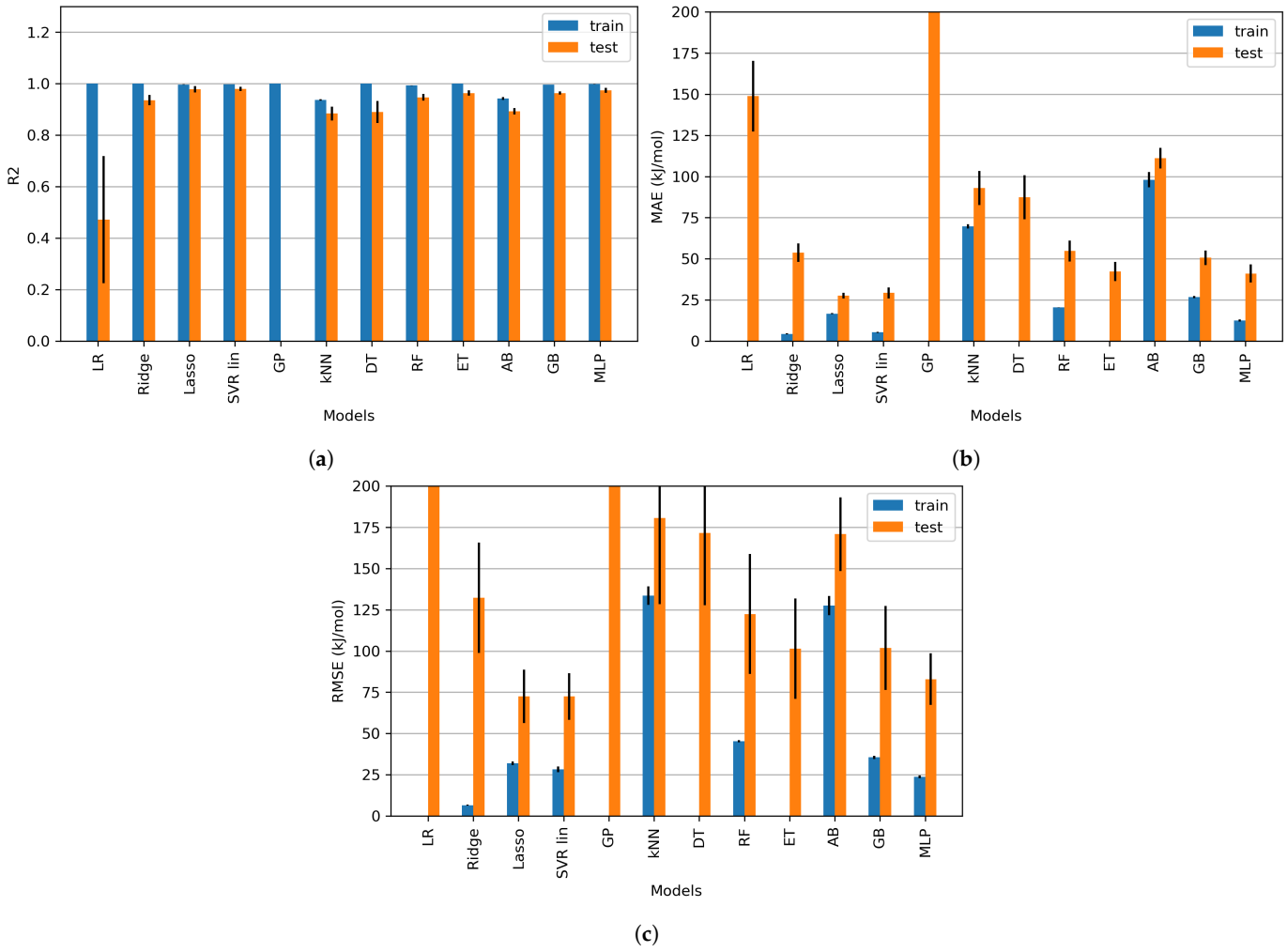
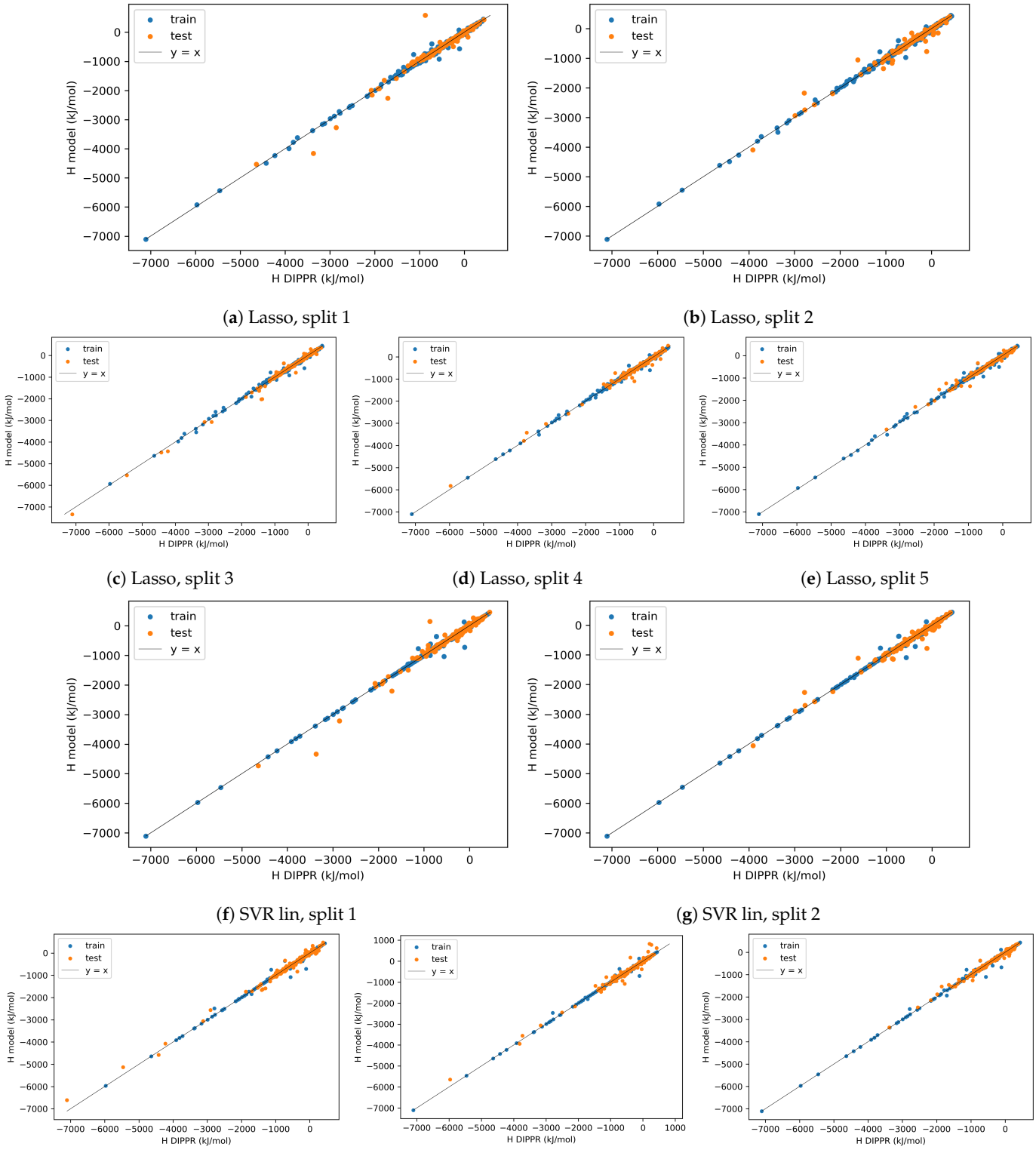


Figure 9. Performance of the different ML models during the preliminary screening for the enthalpy: (**a**) R^2; (**b**) *MAE*; (**c**) *RMSE* (***preprocessing****: default,* ***splitting****: 5-fold external CV,* ***scaling****: standard,* ***dimensionality reduction****: none,* ***HP optimization****: none).*

Similarly, as Ridge and Lasso, SVR lin [98,99] performs better than LR with high-dimensional data. As shown in the objective function of SVR lin (Equation (6), equivalent to Equation (5) with a linear kernel), the left term enables penalizing coefficients to limit overfitting, while the right term controls, via the regularization parameter C, the importance given to the points outside the epsilon tube which surrounds the regression line. Instead of focusing on minimizing the distance between data and model as in LR, Ridge and Lasso, the objective function of SVR lin attempts to minimize the distance between data outside the epsilon tube and the epsilon tube itself. Figure 11 displays the shrinking of the coefficients with Ridge, Lasso and SVR lin methods with respect to the classical LR model. It can be observed that the shrinking effect is more pronounced for Lasso, followed by SVR lin and Ridge, which is consistent with the observed performances and overfitting degree.



(**a**) Lasso, split 1

(**b**) Lasso, split 2

(**c**) Lasso, split 3

(**d**) Lasso, split 4

(**e**) Lasso, split 5

(**f**) SVR lin, split 1

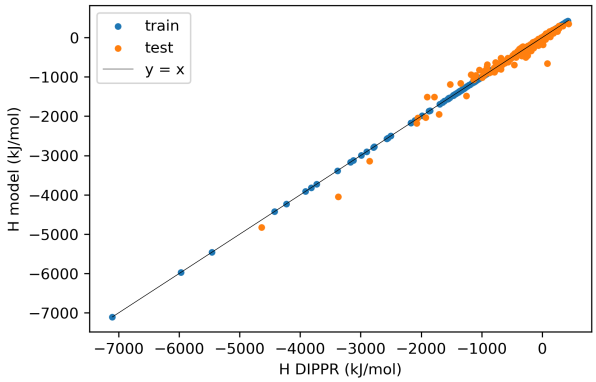
(**g**) SVR lin, split 2

(**h**) SVR lin, split 3

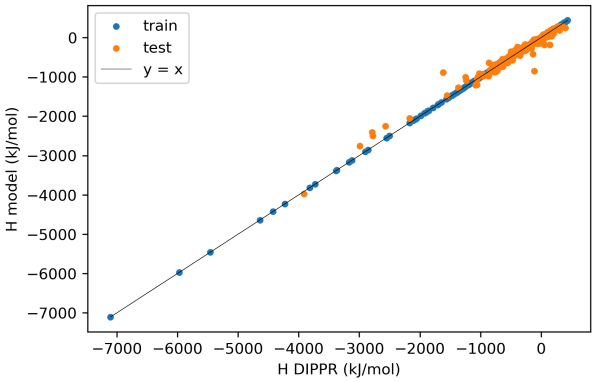
(**i**) SVR lin, split 4

(**j**) SVR lin, split 5

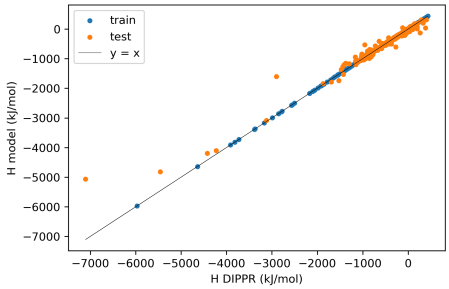
Figure 10. *Cont.*



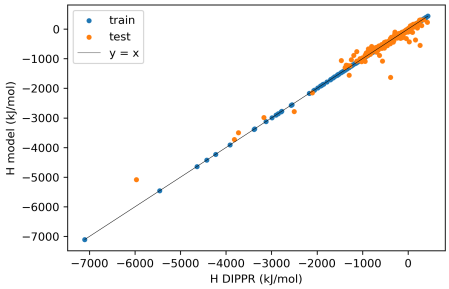
(**k**) ET, split 1



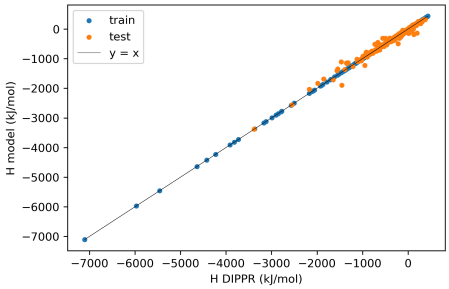
(**l**) ET, split 2



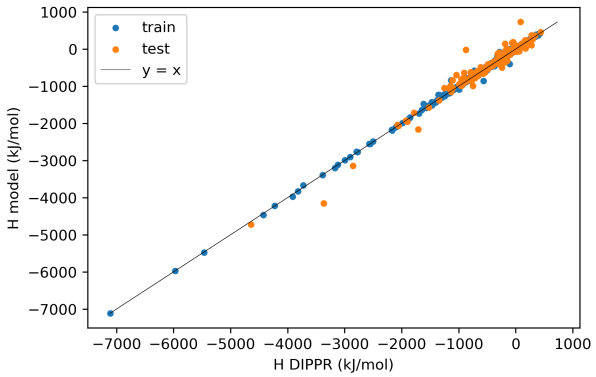
(**m**) ET, split 3



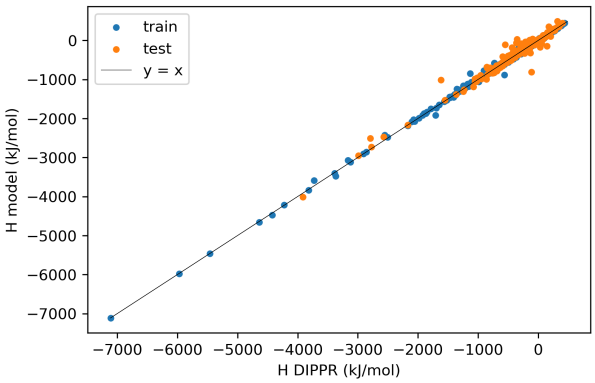
(**n**) ET, split 4



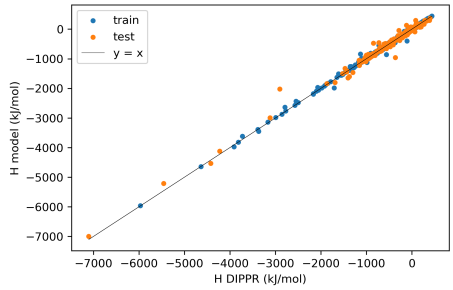
(**o**) ET, split 5



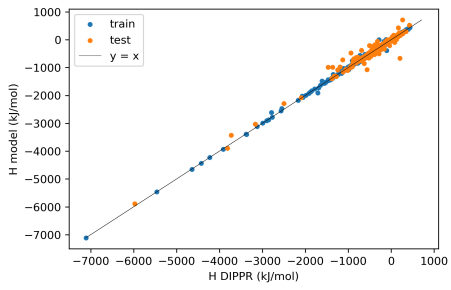
(**p**) MLP, split 1



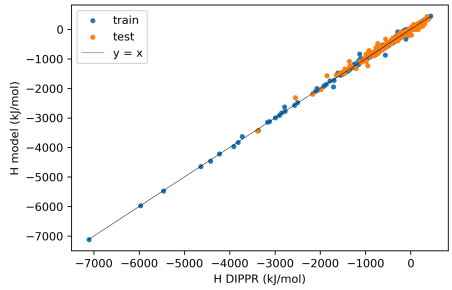
(**q**) MLP, split 2



(**r**) MLP, split 3



(**s**) MLP, split 4



(**t**) MLP, split 5

Figure 10. Parity plots of the selected ML models during the preliminary screening, for different splits, for the enthalpy (***preprocessing***: *default*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

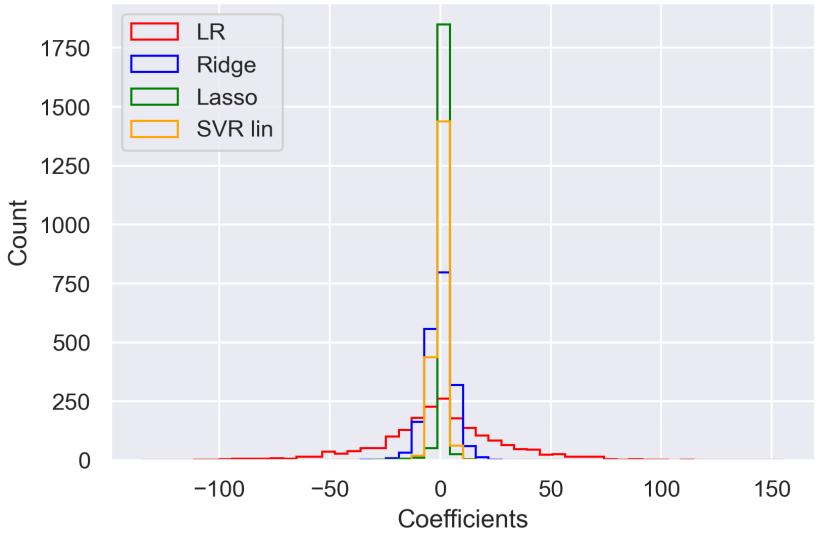


Figure 11. Distribution of the coefficients in various linear regression models during the preliminary screening, for split 1, for the enthalpy (***preprocessing***: *default*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

Objective functions:

- Linear regression:

$$min_{w,b}\|Xw + b - y\|_2^2 \tag{2}$$

- Ridge:

$$min_{w,b}\|Xw + b - y\|_2^2 + \alpha\|w\|_2^2 \tag{3}$$

- Lasso:

$$min_{w,b}\frac{1}{2n}\|Xw + b - y\|_2^2 + \alpha\|w\|_1 \tag{4}$$

- SVR and SVR lin:

$$SVR : min_{w,b}\frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}(\zeta_i + \zeta_i^*) \tag{5}$$

$$SVR_{lin} : min_{w,b}\frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} max(0, |Xw + b - y| - \epsilon) \tag{6}$$

subject to, for $i = 1 \ldots n$:

$$\{y_i - wx_i - b \leq \epsilon + \zeta_i;\ wx_i - b - y_i \leq \epsilon + \zeta_i^*;\ \zeta_i, \zeta_i^* \geq 0\} \tag{7}$$

in the above, n is the number of training molecules, y is the vector of observed values, α and C are regularization parameters, ϵ is the radius of the ϵ-tube surrounding the regression line and ζ_i, ζ_i^* are the distances between the ϵ-tube and the points outside of it.

The results of GP show a perfect fit to the training data but the model is completely unable to adapt to the test data, resulting in excessive overfitting (R^2 train = 1, R^2 test = 0). This could be attributed to the principle of GP which is based on the prediction of a posterior distribution over functions from a prior distribution over functions and the available training data. Predictions are typically accompanied by uncertainties, in contrast to other regression models, which is an important comparative advantage of GP. These uncertainties are more or less important depending on whether the training data cover the feature space around the new test data. However, in high-dimensional spaces, points eventually become

equidistant [100,101] and the feature space contains many empty regions. In certain cases, a pertinent choice of the prior distribution, on the basis of existing knowledge on the behavior of the response with respect to the features has been proven helpful in improving the prediction performance [102–104]. However, such knowledge is not available in the present study.

Likewise, DT is also a method that displays overfitting in this problem. The principle of DT is based on the sequential partition of the training data (root node) into continuously smaller groups, according to a set of decision rules (internal nodes or branches), until the minimum required number of samples for the final nodes (leaf nodes) is reached. However, the construction of a DT can be very sensitive to small variations in the training data and result in overly-complex trees [88]. This phenomenon can be amplified in the presence of a large number of features, which is the case here, thus leading the model to learn rules that are too complex to be generalized to new data.

Different ensemble methods based on DT, namely RF, ET, AB and GB, are also tested to assess whether the combination of the predictions of a large number of DT can improve the generalization performance of the model. As shown in Figure 9, these performances are effectively improved when using these ensemble methods instead of a single DT, except for AB. Ensemble methods can be categorized into bagging (i.e., RF, ET) and boosting (i.e., AB, GB) methods. "Bagging" refers to the strategy of training in parallel several strong estimators (e.g., large DT that present eventual overfitting) on a bootstrap sample of the training data. The individual predictions are then combined to give one final prediction, in the form of an average value, thus reducing the variance of the overall model. In "boosting", several weak estimators (e.g., small DT accompanied by eventual underfitting) are trained sequentially with, at each iteration, a new estimator trained by considering the errors of the previous one. The idea here is that each new estimator attempts to correct the errors made by the previous one, resulting in less overall bias.

The different performances observed for the tested ensemble models can be explained by the slight variations in their mechanisms. For bagging, the difference between RF and ET lies in the method used to compute the splits: RF selects the optimum split while ET selects it at random to further reduce the variance in comparison to RF. As for boosting, GB seems to perform better than AB and this can be attributed to different reasons. While no weighing is applied to the samples in GB, AB increases (resp. decreases) the weights of the training samples with the highest (resp. lowest) errors after each iteration. Additionally, to make the final prediction, each individual estimator in AB is weighted based on its error, while an identical weight is applied to the estimators of GB. These two differences result in a lower generalization capacity for AB to new data, as the most problematic training samples benefit from more attention during the different iterations [88].

The high dimensionality and the problem of the significance of the distances between points may also be the source of the poor performance of kNN, as can be seen in Figure 9. kNN is a distance-based method and its predictions for a new data point are based on the mean property of the k-nearest training neighbors of this point. The distance can be measured via different distance metrics, such as the Euclidean distance. However, when this calculation is carried out over a large number of dimensions, the average distance between points becomes of lower significance and, as such, the concept of "nearest neighbors" becomes weaker. Finally, MLP performs slightly better than all ML models except Lasso and SVR lin. This good performance could be explained by the well-known ability of MLP to approximate any linear/nonlinear function through the complexity of its inner structure.

This first screening only provides a general idea of the most adapted ML techniques to the problem in question but remains bound to the choice of the default values of the HPs of each method. In fact, the HPs of some ML models, such as the selection of kernels in GP and SVR and the number of neurons and hidden layers in MLP, can sometimes display a significant impact on their performance. However, it becomes virtually impractical to consider the implementation of a HP optimization process within a screening step of numerous ML techniques, as this will severely increase the development time and

complexify the selection process. Accordingly, the strategy that has been adopted in the present study consists of sequencing this initial screening with a preprocessing step, a dimensionality reduction step and a HP optimization one only for a selection of the most performing ML models (i.e., as identified through the initial screening step).

The need for an investigation of the effect of a dimensionality reduction step stems from the observed overfitting behavior in Figure 9 for the tested ML models, coupled with the identified performance improvement by the regularization, as employed within the different linear models. Besides, the very nature of the problem includes the manipulation of a large number of descriptors as features of the developed models, for which prior understanding is very limited, renders the dimensionality reduction step a rather obvious necessity in terms of improving both model performance and eventual subsequent interpretability. Finally, another factor that acts in favor of overfitting, in combination with the above, is the consideration of a large diversity of molecules, as evidenced by the respective error bars of the different splits.

It is worth noting that, already from this initial model screening, it seems as if linear models (i.e., Lasso and SVR lin) are sufficient to map the link between molecular descriptors and the enthalpy. This emphasizes that the use of nonlinear and complex ML models is not always necessary since, depending on the problem characteristics, they might display a poorer performance than simpler linear models. Here, the good performance of some linear models is quite intuitive as they display very similar characteristics to the classical GC methods. One of the most popular GC methods for its accuracy, reliability and wide applicability to large and complex molecules, is the one proposed by Marrero and Gani [44]. It is described by Equation (8) which linearly estimates a given property based on first, second and third order molecular groups. First order groups consist of a large set of basic groups, allowing them to represent a wide variety of organic compounds. Higher order groups are included to refine the structural information of molecular groups by accounting for proximity effects and isomer differentiation, thus enlarging GC applicability to more complex molecules. C_i, D_j and O_k represent the contribution of the first, second and third order groups, respectively, occurring N_i, M_j and E_k times, respectively:

$$\widehat{y} = \sum_i N_i C_i + \sum_j M_j D_j + \sum_k E_k O_k \quad (8)$$

3.1.2. Comparison of Data Scaling and Data Splitting Methods

As for data splitting and scaling methods, their effects are, respectively, described in Figures 12 and 13. In particular, 5-fold and 10-fold external CV are compared in terms of train MAE, test MAE and training time. Train MAE values are very similar for all models, except for LR, for both 5-fold and 10-fold external CV. Test MAE values are slightly better for 10-fold external CV for most models, which could be explained by the larger size of the training samples. In addition, the 5-fold external CV naturally requires lower computation times than the 10-fold external CV, due to the lower number of model training passes. Note, that the training time of the different ML models can serve as a factor in the selection of the ML model, depending on the problem requirements. For example, among the ensemble models with close performances (such as RF, ET and GB), ET is the most interesting in terms of computation time, due to the parallel training of several trees and the random splits of the data.

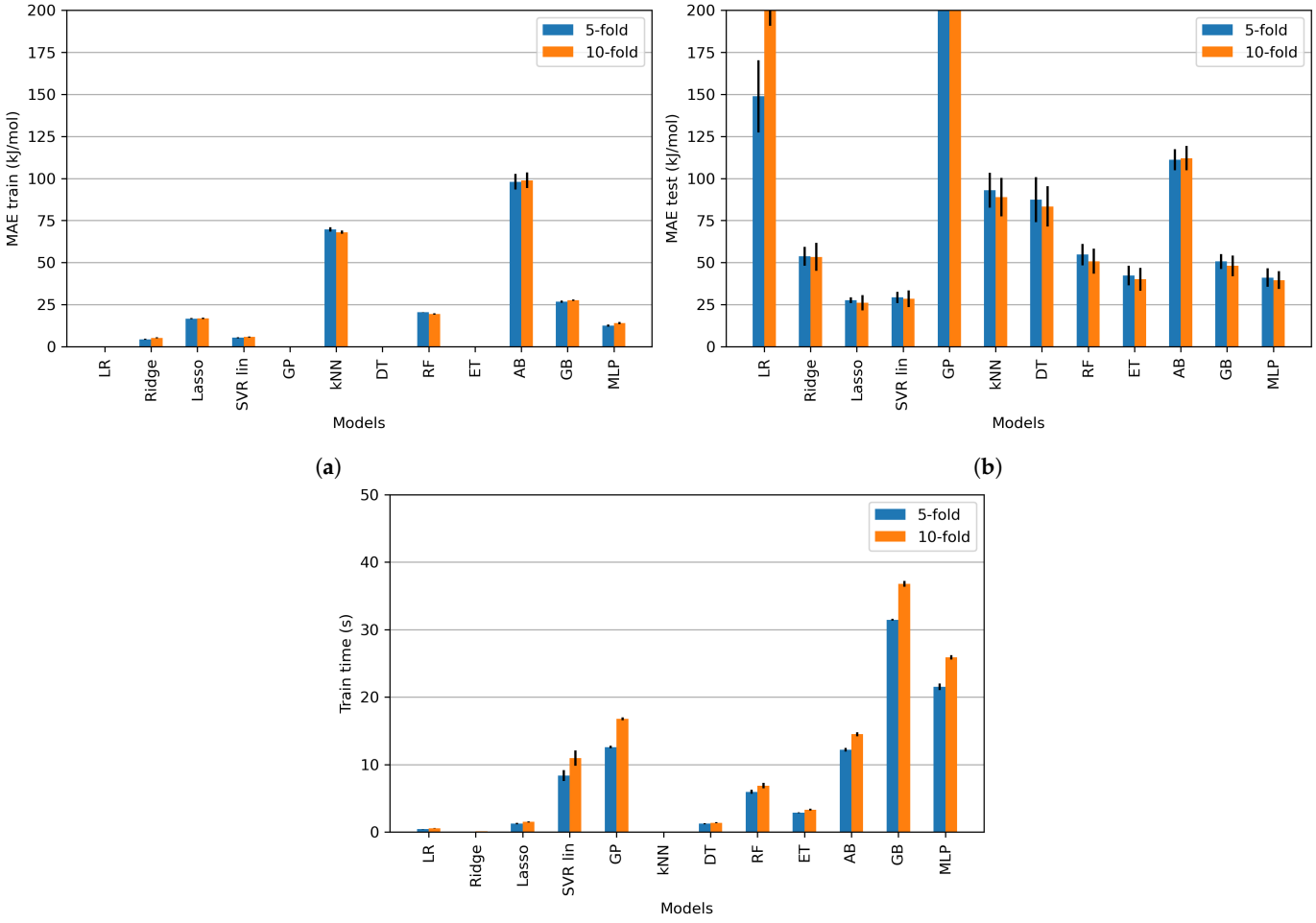


Figure 12. Effect of the value of k', for the external CV, on the (**a**) train *MAE* (**b**), test *MAE*, (**c**) training time, of the different ML models during the preliminary screening for the enthalpy (***preprocessing***: *default*, ***splitting***: *5 and 10-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

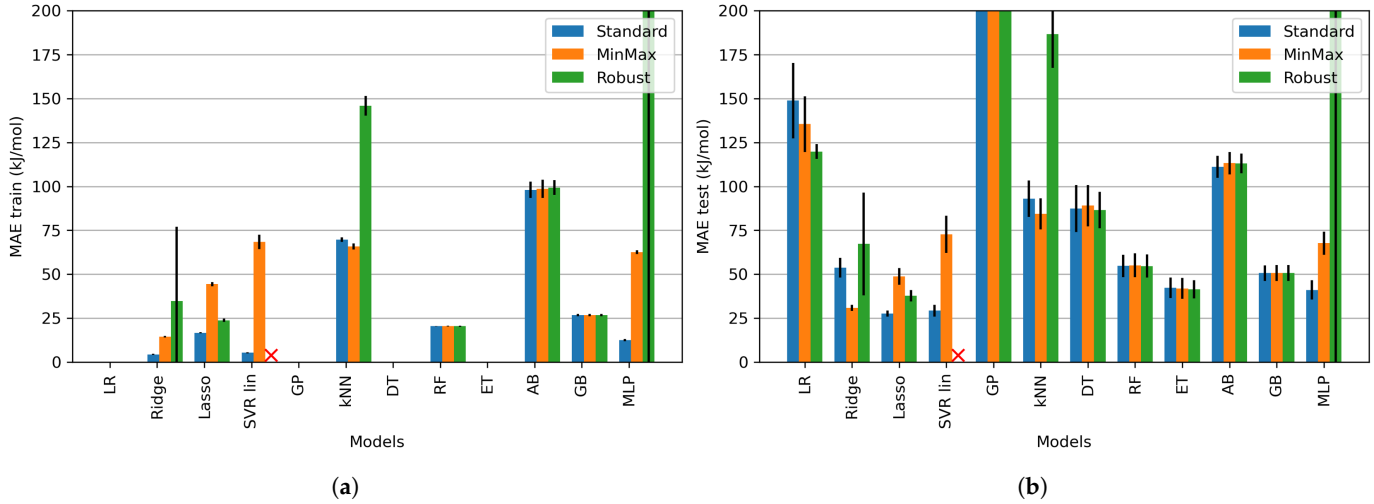


Figure 13. Effect of the data scaling technique on the (**a**) train *MAE* (**b**), test *MAE*, of the different ML models during the preliminary screening for the enthalpy (***preprocessing***: *default*, ***splitting***: *5-fold external CV*, ***scaling***: *standard/min-max/robust*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*). *N.B. Robust scaler did not work with the SVR lin method (cf. red crosses).*

Concerning data scaling, the results in Figure 13 show that the method used can impact more or less the performance (train and/or test) of ML models. On the one hand, single and ensemble DTs show no variations along the tested scaling methods since, at each decision node, a DT finds the best split of the data according to a given descriptor (ignoring the other descriptors), by identifying the threshold minimizing the error. On the other hand, the tested linear models (i.e., LR, Ridge, Lasso, SVR lin), as well as kNN, GP and MLP are more sensitive to scaling. kNN predictions are based on similarity/distance measurements, hence their performance is affected by variations in the value range of the descriptors. The default solver of MLP is based on gradient descent, the range of the descriptors might also influence the gradient descent steps and convergence. The calculation of the information matrix that will be employed within LR for the estimation of the coefficient values will also be affected by the value range of the descriptors. Similar hypotheses, concerning the parametric estimation processes within each method and their sensitivity to the range of the descriptor values can be adopted to explain the observed variations for the rest of the ML models. More generally, robust scaler seems to display the highest MAE across the different techniques, presumably due to the composition of the data set and was thus considered as the least adapted for this study.

Similar results and conclusions are obtained for the entropy for the quick screening of ML models with default preprocessing options and without dimensionality reduction, as well as for the study of the effects of data scaling and splitting (cf. Supplementary Materials). On the basis of the results of this first screening, the configuration presented in Table 6 was selected to further analyze the data preprocessing and dimensionality reduction methods. The best performing ML models from different categories (linear/nonlinear, ensemble/neural network ...) were chosen, including Lasso, SVR lin, ET and MLP. A standard scaler was selected for the scaling of the data, as it displayed the lowest generalization errors in the preliminary tests for the selected models. In addition, as similar performances were obtained for the 5-fold and 10-fold external CV, the former was kept due to its shorter computation time. Finally, MAE was selected as a performance metric due to the importance of the error measurement in thermodynamic property prediction models and applications.

Table 6. Configurations selected for the study of the effects of data preprocessing and dimensionality reduction, and for HP optimization.

Data Scaling	Data Splitting	ML Models	Performance Metrics
Standard	5-fold external CV	Lasso SVR lin ET MLP	MAE

3.2. Effect of Data Preprocessing

Data preprocessing is composed of three stages, namely the elimination of Desc-MVs, the elimination of descriptors with low variance and the elimination of correlated descriptors. The effect of each step will be analyzed sequentially, with the previously selected configuration in Table 6 starting with the default preprocessing options in Table 3. The effects of data preprocessing are demonstrated here for the Lasso model and the enthalpy. The results obtained for the other selected ML models and for the entropy are provided in the Supplementary Materials.

The first step of data preprocessing is the elimination of Desc-MVs since the consideration of a wide diversity of molecules effectively creates groups of Desc-MVs for some descriptors and for some families of molecules. The effects of the three elimination algorithms (cf. Section 2.3) are displayed in Table 7. In the present problem, the alternating elimination algorithm seems to provide a good compromise between the number of remaining molecules, the number of remaining descriptors and the overall model performance. The elimination 'by row' results in better performance but for a significantly

reduced number of molecules, restricting the applicability domain of the developed model. Inversely, the elimination 'by column' removes a significant amount of information on the molecular structure, leading to molecules that can no longer be differentiated on the basis of the remaining descriptors (i.e., molecule duplicates). The retained method for the elimination of Desc-MVs was, therefore, the alternating elimination algorithm.

Table 7. Effect of the algorithms for the elimination of Desc-MVs on the data set size and Lasso model test *MAE* for the enthalpy (***preprocessing***: *default*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

Elimination Procedure	Data Set with Desc-MVs	Data Set without Desc-MVs	Data Set after Preprocessing	*MAE* Train (kJ/mol)	*MAE* Test (kJ/mol)
Alg.1: by row	1903 × 5666 *mol.* *desc.*	236 × 5666 *0 duplicates*	236 × 1378	7.6 $_{\pm 0.4}$	20.4 $_{\pm 6.4}$
Alg.2: by column	1903 × 5666	1903 × 2855 *73 duplicates*	1903 × 988	21.1 $_{\pm 1.0}$	32.9 $_{\pm 5.9}$
Alg.3: alternating row or column	1903 × 5666	1785 × 5531 *0 duplicates*	1785 × 1961	16.7 $_{\pm 0.4}$	27.6 $_{\pm 1.7}$

mol.: molecules. *desc.*: descriptors. Blue, orange and red colors represent limited, moderate, important information loss, respectively. In column 3, the amount of duplicated rows is indicated in italics. In columns 5 and 6, the standard deviation over the different splits is provided in subscript.

The second step consists of the elimination of descriptors with low variance as they have no influence on the target property. Figure 14a shows the effect of different variance thresholds on the number of remaining descriptors after the elimination of descriptors with low variance and after complete preprocessing. The resulting test *MAE* is also presented to facilitate the choice of the threshold value. By increasing the latter, the number of remaining descriptors naturally decreases inducing a loss of information and an increase in the value of *MAE* for the test data. Accordingly, the value of 0.0001 was chosen to limit the loss of molecular information, while keeping the *MAE* value at its lower range. Note, for the case of the complete preprocessing, shown in Figure 14a, the value of the correlation coefficient was set to 0.95 by default. Qualitatively, the trend of the corresponding curve is similar to other values of the correlation coefficient. Quantitatively, a higher (lower) coefficient value will displace the curve downwards (upwards), as shown in Figure 14b, which illustrates the effect of the coefficient value during the final step of the data preprocessing, namely that of the elimination of linearly correlated descriptors. Note that, in Figure 14b, the value of the low variance threshold is the one previously selected (0.0001). The value that was finally retained for the correlation coefficient is 0.98, for identical reasons as for the choice of the low variance threshold.

Similar results and conclusions were obtained for the entropy regarding the effects of data preprocessing. In the rest of this article, the selected preprocessing options of this section (i.e., elimination of the Desc-MVs by alternating row and column, elimination of descriptors with variance $\leq$0.0001, elimination of descriptors with correlation coefficient value $\geq$0.98) are referred to as the 'final' preprocessing options for both predicted thermodynamic properties. The summary of selected preprocessing options is presented in Table 8.

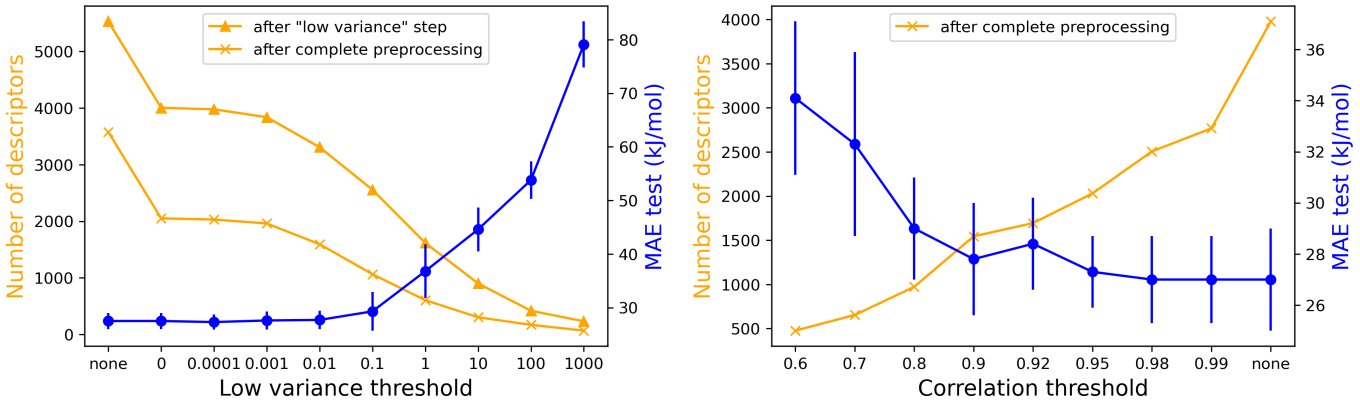
3.3. Effect of the Dimensionality Reduction

The number of descriptors is still relatively high after data preprocessing (i.e., 2506 descriptors for the enthalpy with the final options), and dimensionality reduction methods are investigated to further enhance interpretability, performance and computation time of the ML models. In particular, the effects of different feature selection methods (i.e., two filter methods, two wrapper methods and three embedded methods) and of one feature extraction method (i.e., PCA) are compared with the reference case in which no dimensionality reduction is performed. For a fair comparison of the effects of the feature

selection methods, they are all employed under a common objective of reducing the feature space to an exact number of 100 descriptors. On the other hand, the principal components (PCs) selected by PCA correspond to 95% of the variance of the data. To prevent data leakage, dimensionality reduction methods are fitted on the training data and applied to all the data for each split of the 5-fold external CV, thus providing, at the same time, the influence of data splitting.

Table 8. Summary of the final preprocessing options.

Preprocessing Step	Final
Elimination of Desc-MVs	Alternating row or column
Elimination of descriptors with low variance	0.0001
Elimination of correlated descriptors	0.98



(**a**) (**b**)

Figure 14. Effect of the value of (**a**) the variance threshold (**b**) the correlation coefficient threshold, on the number of retained descriptors and Lasso model test *MAE* for the enthalpy (***preprocessing***: *default for* (***a***) *and default with low variance threshold = 0.0001 for* (***b***), ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

The results are presented for the enthalpy in Tables 9 and 10 as an average of the different splits. The displayed computation time is the one for fitting the dimensionality reduction methods for each split. Wrapper methods are the most time-consuming as they consist of a more comprehensive search of the optimal subset of descriptors. These methods are based on Lasso as it displayed both good performance and low computation time in Section 3.1. The computation time of the GA method is mainly dependent on the number of generations, which was set here to 5000, keeping in mind that a different value would affect not only the computation time but also the performance of the model. Note also that a gain is expected in the computation time of the subsequent ML training step that should compensate in part the additional time investment to this dimensionality reduction step (i.e., besides the aforementioned envisioned benefits of improved interpretability and performance).

In terms of performances, the test *MAE* values of previously identified well-performing ML models (i.e., Lasso, SVR lin, ET and MLP) are compared among the different dimensionality reduction methods. To aid in the legibility, the values that are noted in blue color, in Table 9, correspond to test *MAE* values that are either lower or within a difference $\leq$0.5 kJ/mol, compared to the respective reference case values (i.e., without dimensionality reduction). In the same sense, test *MAE* values that are higher by a difference that is $\leq$5 or $>$5 kJ/mol, compared to the reference case, are marked in orange and red, respectively.

Table 9. Effect of the different dimensionality reduction methods on the test *MAE* of the selected ML models for the enthalpy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *different methods*, ***HP optimization***: *none*).

Dimensionality Reduction Method	Nb of Desc.	Time/ Split (s)	*MAE* Test (kJ/mol)				Nb of Pairwise Correlations $\geq$ 0.9	Nb of Desc. with Variance $\leq$ 0.01
			Lasso	SVR lin	ET	MLP		
None (reference case)	2506	0	27.0 $\pm$1.7	28.6 $\pm$3.6	42.6 $\pm$6.6	37.9 $\pm$5.2	4473	512
Filter-Pearson	100	19.4	61.6 $\pm$2.5	75.0 $\pm$5.7	56.7 $\pm$6.1	116.0 $\pm$3.6	124	2
Filter-MI	100	18.7	55.8 $\pm$3.6	62.5 $\pm$7.8	43.8 $\pm$7.8	90.8 $\pm$9.8	72	4
Wrapper-SFS Lasso	100	7795	31.1 $\pm$2.4	34.9 $\pm$3.3	42.9 $\pm$4.8	84.6 $\pm$3.7	3	21
Wrapper-GA Lasso	100	49573	24.2 $\pm$4.0	31.0 $\pm$5.0	43.8 $\pm$5.2	76.1 $\pm$9.7	5	26
Embedded-Lasso	100	1.5	29.0 $\pm$1.4	29.0 $\pm$2.4	38.8 $\pm$5.5	88.7 $\pm$9.9	14	24
Embedded-SVR lin	100	7.0	39.8 $\pm$5.1	40.0 $\pm$5.6	41.1 $\pm$6.1	84.1 $\pm$4.2	34	18
Embedded-ET	100	3.7	50.3 $\pm$4.1	51.2 $\pm$4.0	41.4 $\pm$5.9	85.5 $\pm$9.0	46	4
PCA 95% (261–265 PC)	2506	2.9	37.3 $\pm$3.3	34.2 $\pm$2.7	76.8 $\pm$12.1	38.0 $\pm$2.2	-	-

Table 10. Top five descriptor categories identified by the different dimensionality reduction methods for the enthalpy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *different methods*, ***HP optimization***: *none*). *The percentages correspond to the proportion of a descriptor category among the descriptors obtained with each method.*

Dimensionality Reduction Method	Top 5 Descriptor Categories				
None (reference case)	25 13.0%	19 8.5%	8 7.2%	30 6.3%	17 6.3%
Filter-Pearson	8 24.4%	3 16.6%	16 15.4%	19 13.2%	7 8.2%
Filter-MI	8 21.4%	7 19.2%	3 13.0%	11 9.8%	27 8.6%
Wrapper-SFS Lasso	25 16.8%	23 10.6%	22 9.0%	21 6.6%	17 6.0%
Wrapper-GA Lasso	25 22.6%	23 10.4%	22 9.2%	21 8.4%	10 8.2%
Embedded-Lasso	25 20.8%	22 8.4%	7 7.6%	10 7.4%	23 7.4%
Embedded-SVR lin	25 32.2%	23 9.8%	10 8.8%	22 7.8%	1 7.2%
Embedded-ET	12 16.0%	8 13.6%	7 13.4%	3 8.6%	11 7.2%
PCA 95% (261–265 PC)	25 13.0%	19 8.5%	8 7.2%	30 6.3%	17 6.3%

Accordingly, one can directly conclude from the results of Table 9 that a reduced number of 100 descriptors is sufficient to provide better or similar results to the reference case of 2506 descriptors. This is especially observed with the wrapper methods and the Lasso-based embedded method and for the ML models of Lasso, SVR lin and ET. The wrapper-GA Lasso method performs better than the wrapper-SFS Lasso model, which might be due to the lower flexibility of the latter in terms of the treatment of descriptors with respect to the former. In fact, GA has the ability to completely modify the population of individuals (i.e., one individual being represented here by one subset of 100 descriptors), after each generation, while SFS adds descriptors iteratively until reaching the required number of descriptors. This means that, in SFS, descriptors can not be removed once they have been selected, even in the case where they might no longer be interesting after the addition of new ones, which does not apply to GA. As for the Lasso-based embedded method, it internally identifies the subset of the most relevant descriptors during training. Inversely, filter methods result in poorer prediction performances, as the importance of each descriptor is evaluated independently.

From the results, it can also be observed that PCA is not adapted to such highly dimensional problems. Figure 15a,b display the explained variance as a function of the principal components for the enthalpy and the entropy, respectively. In the present case, for both enthalpy and entropy, more than 250 PCs are required to describe 95% of the data variance,

each one being a linear combination of nearly 2500 descriptors. Regarding embedded methods, Lasso outperforms SVR lin and ET, in the sense that it identifies a drastically reduced subset of important descriptors. Indeed, the selected 100 descriptors are the ones that display the highest absolute coefficient values (absolute feature importance values for ET) and Lasso, SVR lin and ET result respectively in a number of 252, 2494 and 2268 non-zero coefficient or feature importance values. The performance of MLP models does not show significant improvement with any of the dimensionality reduction methods, but their performance is very sensitive to HP values and thus, likely to improve with further HP optimization. It should be highlighted here that the results of this dimensionality reduction step are also highly associated with the choices made during the data preprocessing step.

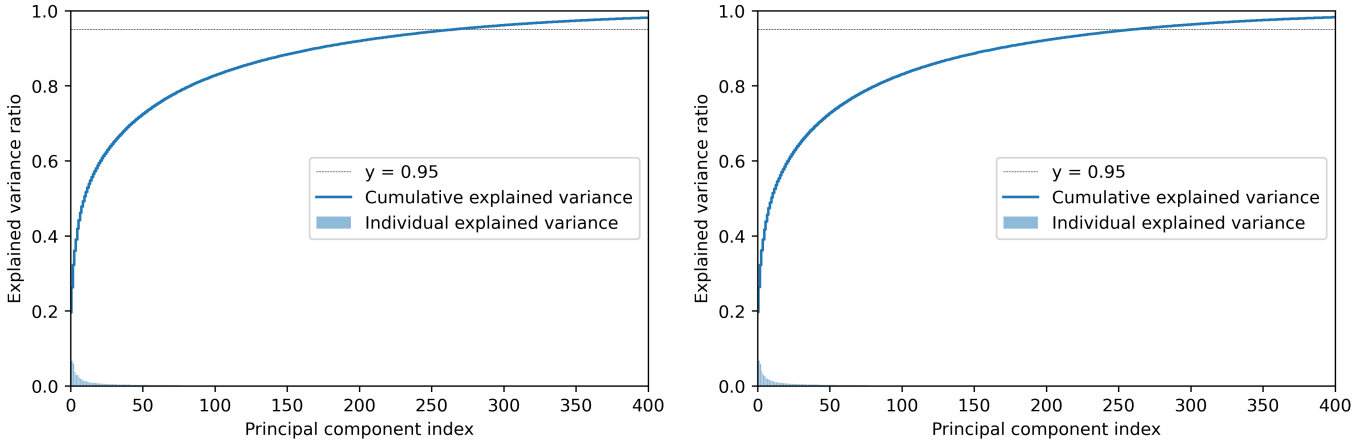


Figure 15. Explained variance as a function of the principal components obtained with PCA for (**a**) the enthalpy and (**b**) the entropy. (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *PCA*, ***HP optimization***: *none*).

Another explanation for the good performances obtained with the two wrapper methods and the Lasso embedded method can be visualized in the last two columns of Table 9. They display respectively the amount of pairwise correlations $\geq$0.9 and the number of descriptors with variance $\leq$0.01 (averaged over the different splits) among the descriptors selected by the different dimensionality reduction methods. This highlights the presence of highly correlated descriptors in the case of filter methods as they treat descriptors independently, thus impacting the performance of the ML models. These filter methods also identify as important only a few descriptors with variance $\leq$0.01 contrary to most of the other dimensionality reduction methods that result in better performance.

Depending on the splits, the 100 descriptors or 95% variance based PCs, obtained with the feature selection and PCA methods, respectively, display significant variability in the final model performance as shown in Table 9. This can be mainly due to the fact that each randomly created split corresponds to a different composition of the training data with respect to the represented chemical families. One of the major drawbacks of using descriptors in this type of study lies in their large amount and in their ad-hoc definition, which makes it particularly tedious to understand the meaning of each individual descriptor and its relevance to the property of interest. However, through this dimensionality reduction procedure, it is possible to eventually identify some categories of descriptors (cf., AlvaDesc categories in Table 2) that are more often represented than others, thus demonstrating their higher relevance to the predicted property.

Among the descriptors identified in this work (cf. Table 10 and Supplementary Materials for the detailed list), on the basis of the three best performing dimensionality reduction methods (i.e., two wrapper methods and one embedded method based on Lasso), 2D descriptors seem to be the most represented ones. More specifically, these include the 2D atom pairs (category 25), atom-centered fragments (cat. 22) and atom-type e-state indices (cat.

23). The two former provide information about the presence/absence/count/topological distance of atom pairs or atom-centered fragments while the latter describes the electronic character and the topological environment of the atoms in a molecule. These identified descriptors are physically consistent with the prediction of the enthalpy of a molecule that is highly dependent on its chemical bonds and environment. At the same time, they are also quite similar to the procedure employed by GC, which decomposes molecules in smaller groups to obtain the global property but also develops certain corrections to account for specific interactions (e.g., interactions between bulky groups about σ bonds in alkanes or about π bonds in alkenes) or geometrical particularities (e.g., the presence of a ring inducing additional strain energy) in more complex molecules (cf. Equation (8) and [39,105–108]). The following categories are also represented at a lower extent and give additional 2D and 3D structural information impacting the enthalpy: 2D matrix-based descriptors (cat. 7), P_VSA-like descriptors (cat. 10), 3D-MoRSE descriptors (cat. 17) and functional group counts (cat. 21). For further information and understanding of the identified descriptor categories, a brief description is provided, for each one of them, in the Supplementary Materials.

A similar analysis can be made for the results of the dimensionality reduction, when it comes to the prediction of the entropy (cf. Tables 11 and 12). The best performing dimensionality reduction methods turn out to be the same as for the enthalpy, namely the two wrapper methods and the Lasso-based embedded method. As for the corresponding most represented categories, they include 2D and 3D descriptors: 2D atom pairs (cat. 25), functional group counts (cat. 21) and CATS 3D descriptors (cat. 30). The presence of the latter is not surprising as the entropy is known to be highly sensitive to the spatial arrangement of atoms in molecules and how restricted are their movements, and CATS 3D descriptors effectively include information about the Euclidean interatomic distance between two given atom types. In particular, entropy is a fingerprint of the number of possible microstates of a species in thermodynamic equilibrium. It is derived from a molecular partition function describing translational energy states, rotational energy levels, electronic states, and vibrational ones. It is also reflecting the presence of symmetries (internal and external ones), and optical isomers. As for the two other descriptor categories, 2D atom pairs and functional group counts, they give information about the arrangement of atoms in molecules and their presence seems in accordance with the procedure employed by GC. With lower importance, 2D matrix-based descriptors (cat. 7), RDF descriptors (cat. 16), atom-centered fragments (cat. 22), atom-type e-state indices (cat. 23) and pharmacophore descriptors (cat. 24) are also identified as being highly relevant.

Table 11. Effect of the different dimensionality reduction methods on the test *MAE* of the selected ML models for the entropy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *different methods*, ***HP optimization***: *none*).

Dimensionality Reduction Method	**Nb of Desc.**	**Time/ Split (s)**	***MAE* Test (J/mol/K)**				**Nb of Pairwise Correlations $\geq$ 0.9**	**Nb of Desc. with Variance $\leq$ 0.01**
			Lasso	**SVR lin**	**ET**	**MLP**		
None (reference case)	2479	0	18.7 $\pm$1.1	24.3 $\pm$2.2	19.6 $\pm$1.4	27.1 $\pm$1.6	4469	487
Filter-Pearson	100	18.1	20.9 $\pm$1.3	18.5 $\pm$1.7	20.3 $\pm$1.3	46.9 $\pm$3.0	909	2
Filter-MI	100	16.3	21.4 $\pm$1.4	19.1 $\pm$1.4	20.0 $\pm$1.6	31.4 $\pm$1.0	514	6
Wrapper-SFS Lasso	100	8294	19.2 $\pm$1.2	18.9 $\pm$1.7	19.6 $\pm$1.9	55.8 $\pm$4.2	15	19
Wrapper-GA Lasso	100	53,315	17.4 $\pm$1.2	18.1 $\pm$1.3	19.6 $\pm$1.5	57.5 $\pm$3.7	9	25
Embedded-Lasso	100	1.4	18.8 $\pm$1.1	18.5 $\pm$1.2	19.7 $\pm$1.5	52.6 $\pm$5.6	21	20
Embedded-SVR lin	100	5.6	24.2 $\pm$2.5	23.5 $\pm$2.8	21.3 $\pm$2.0	53.0 $\pm$4.6	8	17
Embedded-ET	100	3.7	21.6 $\pm$1.9	19.7 $\pm$1.4	20.8 $\pm$2.1	31.0 $\pm$1.7	338	8
PCA 95% (254–260 PCs)	2479	3.0	19.7 $\pm$1.0	18.3 $\pm$1.5	23.0 $\pm$1.3	29.1 $\pm$2.3	-	-

Table 12. Top 5 descriptor categories identified by the different dimensionality reduction methods for the entropy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *different methods*, ***HP optimization***: *none*).

Dimensionality Reduction Method	Top 5 Descriptor Categories				
None (reference case)	25 12.9%	19 8.7%	8 7.2%	17 6.3%	30 6.2%
Filter-Pearson	7 16.2%	16 15.8%	19 15.4%	3 7.0%	14 7.0%
Filter-MI	7 46.8%	14 10.6%	3 8.4%	8 6.2%	1 4.0%
Wrapper-SFS Lasso	25 14.6%	30 10.4%	21 7.0%	7 6.0%	24 5.4%
Wrapper-GA Lasso	25 16.6%	21 10.4%	30 7.0%	24 5.4%	7 5.0%
Embedded-Lasso	25 17.2%	21 9.6%	16 8.0%	30 7.8%	23 6.0%
Embedded-SVR lin	25 15.8%	30 12.0%	16 8.0%	24 7.4%	21 6.4%
Embedded-ET	7 28.2%	8 14.6%	14 11.8%	9 9.8%	19 8.4%
PCA 95% (254–260 PCs)	25 12.9%	19 8.7%	8 7.2%	17 6.3%	30 6.2%

Note, that the final selection of a single dimensionality reduction method is not straightforward and will depend on the problem requirements, often necessitating a compromise between performance, computation time and interpretability. However, the comparison of different dimensionality reduction approaches, as employed in the present work, provides a higher degree of confidence with the identification of the descriptors and, accordingly, of the molecular characteristics that display the highest relevance to the target property.

3.4. Final ML Modeling and HP Optimization

A final ML modeling step is performed here, similarly as in Section 3.1. The pretreatment of the data in this case includes the final preprocessing options and is followed by the dimensionality reduction step using the wrapper-GA Lasso method, as shown previously. This choice is based on the premise that the main interest here is the model performance, despite the increased computation time. Should the computation time be of higher interest, a different dimensionality reduction approach would have been selected (e.g., embedded-Lasso). At this stage (i.e., with a reduced number of descriptors), a screening of the same 12 ML models as in Section 3.1 still identifies the four selected models as being part of the best ones (cf. Supplementary Materials). However, the reduced descriptor space enables to improve significantly the performance of some models such as LR and Ridge. Otherwise, the training time is drastically reduced and a comparison of the different scaling techniques still outputs the standard scaler as the scaling method of choice (cf. Supplementary Materials).

HPs are finally optimized for the four best models of different categories, namely Lasso, SVR lin, ET and MLP. Table 13 presents the different types of HP that are considered for each method, each one accompanied by the range of values within which GridSearch CV performs the screening. The final optimal values that minimize the validation MAE for each split are also reported in the same table. For reasons of completeness, some HPs for which no optimization was pursued (i.e., their values were fixed) are also included in the table. The final ML models with the optimal HP settings are retrained on the training data (external training) and tested on the test data.

The resulting performances and parity plots are shown respectively in Table 14 and Figure 16. From these results, it can be concluded that the employed HP optimization step displays a positive effect, especially on the performance of the MLP. However, this improvement is not enough to outperform Lasso, which remains the overall best-performing model. Note, at this stage, no treatment of possible outlier data took place as this will be the subject of an extensive analysis in the following article. Similar conclusions are obtained for the entropy and the performance and parity plots of the Lasso model with optimized HPs are respectively presented in Table 15 and in Figure 17, the complete results being available in the Supplementary Materials. The latter also provides the coefficient values of the Lasso models for both enthalpy and entropy, to enable further interpretation and eventual implementation of the developed models of this work.

Table 13. HP optimization settings and results for the selected ML models for the enthalpy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *yes*).

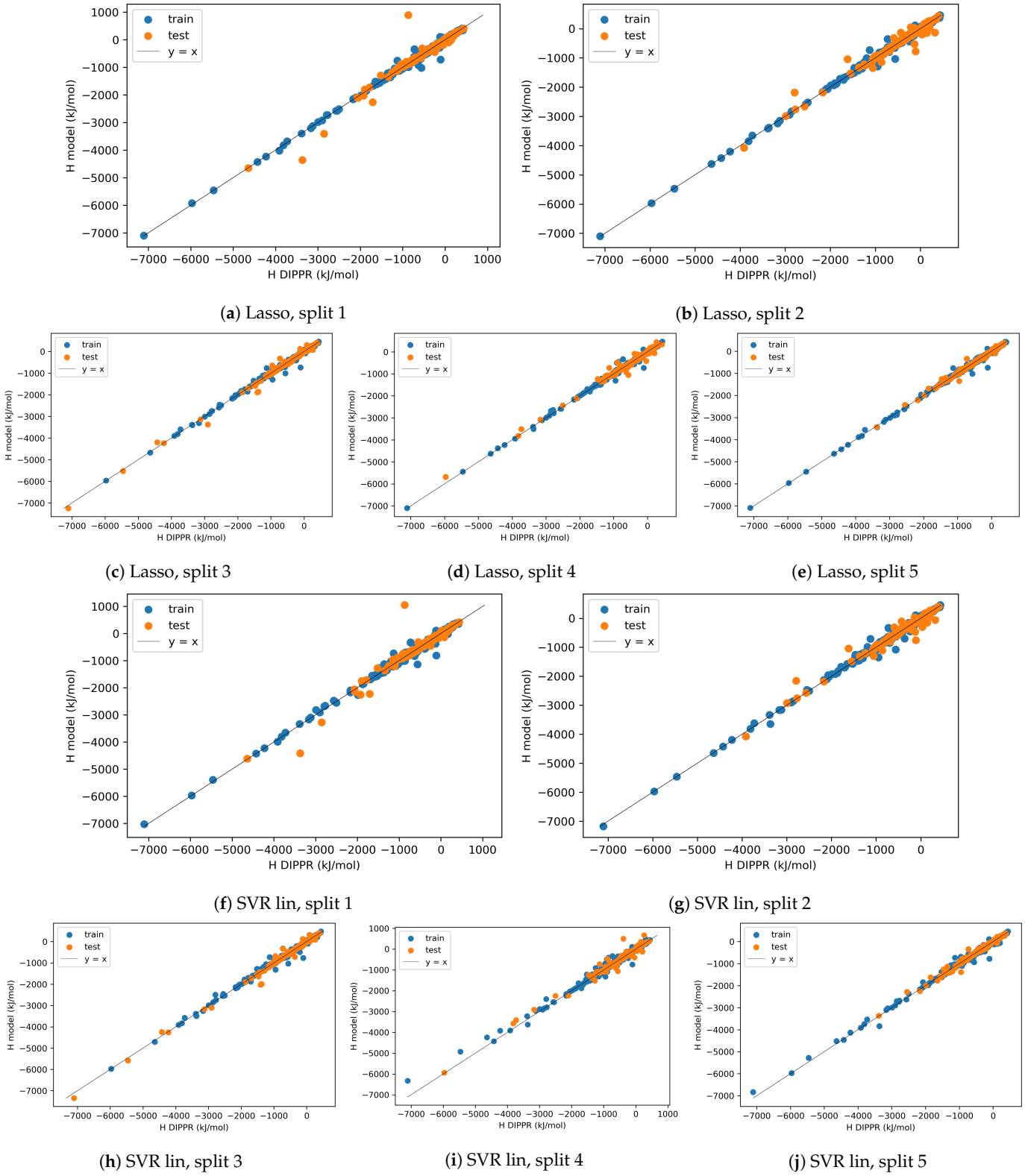
ML Model	HPs	Screening Ranges (Blue = Default Value)	Optimal HP Settings per Split				
			Split 1	Split 2	Split 3	Split 4	Split 5
Lasso	alpha	[0.001, 0.01, 0.1, 0.5, 1, 1.5, 2]	0.1	0.001	0.1	0.5	0.1
SVR lin	kernel	['linear']	linear	linear	linear	linear	linear
	C	[0.1, 0.5, 1, 1.5, 2]	2	2	2	2	2
	epsilon	[0.01, 0.1, 1]	0.1	0.1	1	1	1
ET	n_estimators	[50, 100, 200]	200	200	100	100	200
	max_features	['sqrt', 'log2', None]	None	None	None	None	None
	min_samples_split	[2, 5]	2	2	2	2	2
	min_samples_leaf	[1, 5]	1	1	1	1	1
	max_depth	[10, None]	None	None	None	None	None
	criterion	['absolute error', 'squared error']	squared	squared	squared	squared	squared
MLP	activation	['relu']	relu	relu	relu	relu	relu
	hidden_layer_sizes	1 hidden layer: [(i)], i = 100, 200, 400; 2 hidden layers: [(i, i)], i = 10, 15, 20	(100)	(10,10)	(15,15)	(10,10)	(15,15)
	solver	['adam', 'lbfgs']	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
	learning_rate_init	[0.001, 0.01, 0.1, 0.5]	0.001	0.001	0.001	0.001	0.001
	max_iter	[200, 500]	200	500	500	200	200

Table 14. Performance of the selected ML models with and without HP optimization for the enthalpy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *none/yes*).

Model	Data Set	R^2		*MAE* (kJ/mol)		*RMSE* (kJ/mol)	
		HP Not Opt.	HP Opt.	HP Not Opt.	HP Opt.	HP Not Opt.	HP Opt.
Lasso	Train (internal)	0.995 ±0.001	0.996 ±0.001	15.8 ±0.5	14.6 ±0.3	35.9 ±1.4	33.7 ±1.5
	Validation	0.987 ±0.002	0.989 ±0.002	24.8 ±1.5	22.3 ±0.7	52.2 ±3.5	47.8 ±2.9
	Train (external)	0.995 ±0.001	0.996 ±0.001	15.5 ±0.5	14.6 ±0.3	36.9 ±1.5	34.6 ±1.6
	Test	0.978 ±0.016	0.976 ±0.019	24.2 ±4.0	25.1 ±4.1	70.8 ±23.1	74.2 ±25.9
SVR lin	Train (internal)	0.987 ±0.005	0.993 ±0.002	23.1 ±3.3	17.9 ±1.6	58.2 ±11.9	44.6 ±6.0
	Validation	0.975 ±0.008	0.984 ±0.006	35.6 ±5.3	27.8 ±2.9	75.6 ±15.9	60.1 ±9.9
	Train (external)	0.990 ±0.004	0.993 ±0.002	21.4 ±2.5	17.3 ±1.6	53.5 ±9.6	43.7 ±5.8
	Test	0.968 ±0.023	0.971 ±0.021	31.0 ±5.0	27.8 ±4.5	85.8 ±26.8	82.4 ±26.3
ET	Train (internal)	1.000 ±0.000	1.000 ±0.000	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
	Validation	0.933 ±0.006	0.933 ±0.006	61.6 ±4.9	61.3 ±4.6	114.5 ±9.3	114.4 ±9.2
	Train (external)	1.000 ±0.000	1.000 ±0.000	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
	Test	0.955 ±0.014	0.955 ±0.014	43.8 ±5.2	43.6 ±5.5	112.3 ±36.1	112.3 ±36.7
MLP	Train (internal)	0.955 ±0.006	0.998 ±0.001	79.7 ±7.2	11.8 ±3.7	112.2 ±10.5	20.1 ±6.5
	Validation	0.764 ±0.016	0.964 ±0.009	125.9 ±6.7	42.5 ±2.7	197.0 ±11.9	81.3 ±6.6
	Train (external)	0.968 ±0.005	0.999 ±0.000	65.2 ±8.1	10.3 ±2.4	95.3 ±10.9	17.7 ±3.8
	Test	0.943 ±0.025	0.976 ±0.008	76.1 ±9.7	34.9 ±2.5	117.6 ±12.9	78.3 ±10.9

Table 15. Performance of Lasso model with HP optimization for the entropy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *yes*).

Model	Data Set	R^2	*MAE* (J/mol/K)	*RMSE* (J/mol/K)
Lasso	Train (internal)	0.982 ±0.001	13.8 ±0.4	27.6 ±0.9
	Validation	0.966 ±0.005	17.5 ±0.6	34.5 ±1.8
	Train (external)	0.982 ±0.001	13.7 ±0.4	28.0 ±0.9
	Test	0.968 ±0.008	17.9 ±1.2	36.2 ±4.3



(**a**) Lasso, split 1

(**b**) Lasso, split 2

(**c**) Lasso, split 3

(**d**) Lasso, split 4

(**e**) Lasso, split 5

(**f**) SVR lin, split 1

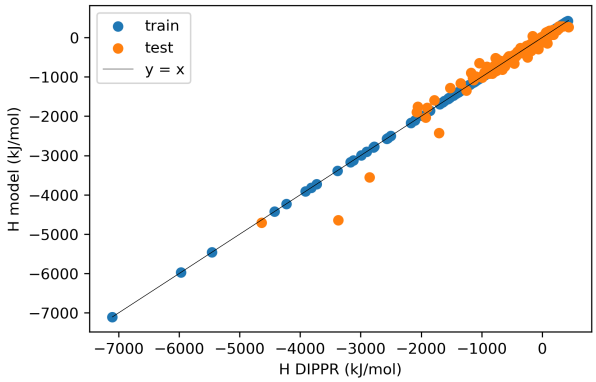
(**g**) SVR lin, split 2

(**h**) SVR lin, split 3

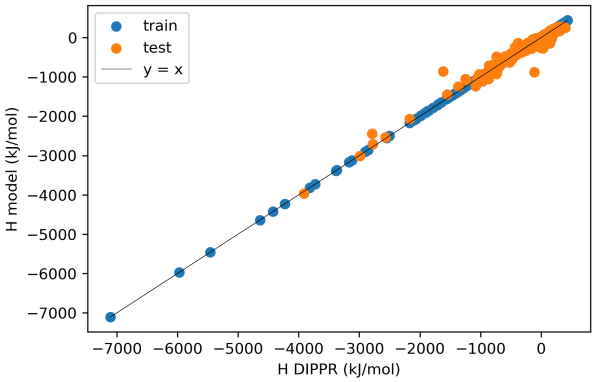
(**i**) SVR lin, split 4

(**j**) SVR lin, split 5

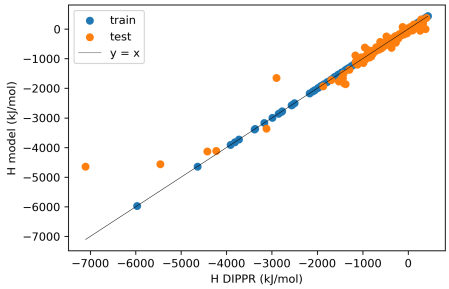
Figure 16. *Cont.*



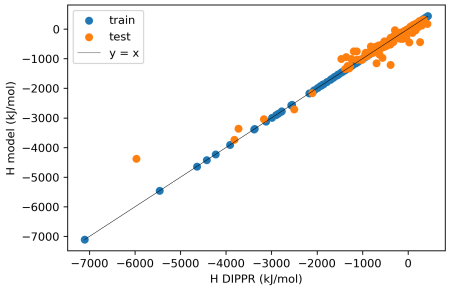
(**k**) ET, split 1



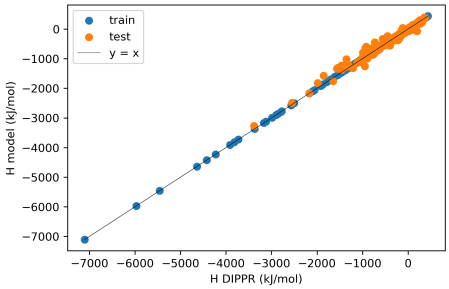
(**l**) ET, split 2



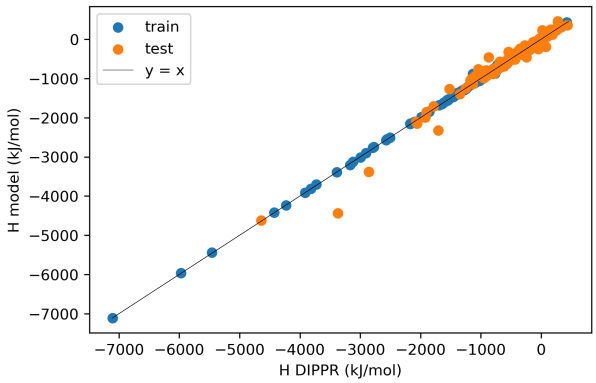
(**m**) ET, split 3



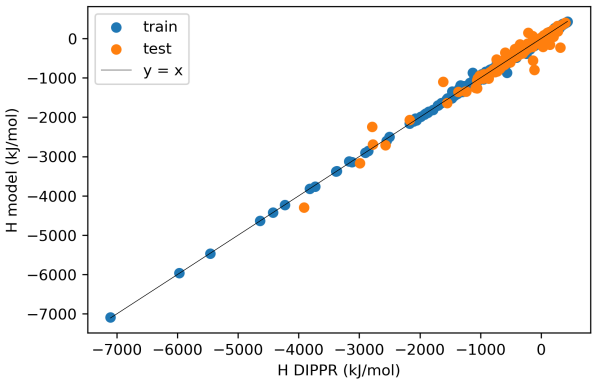
(**n**) ET, split 4



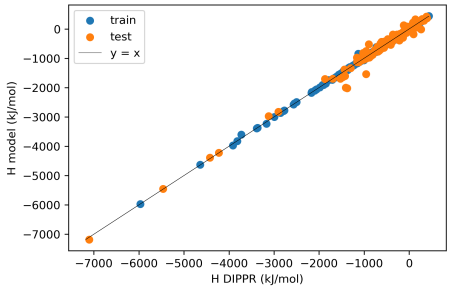
(**o**) ET, split 5



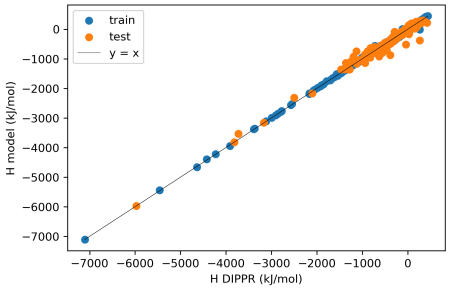
(**p**) MLP, split 1



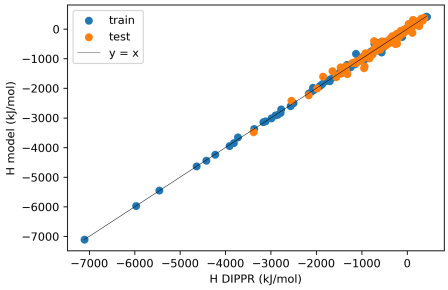
(**q**) MLP, split 2



(**r**) MLP, split 3

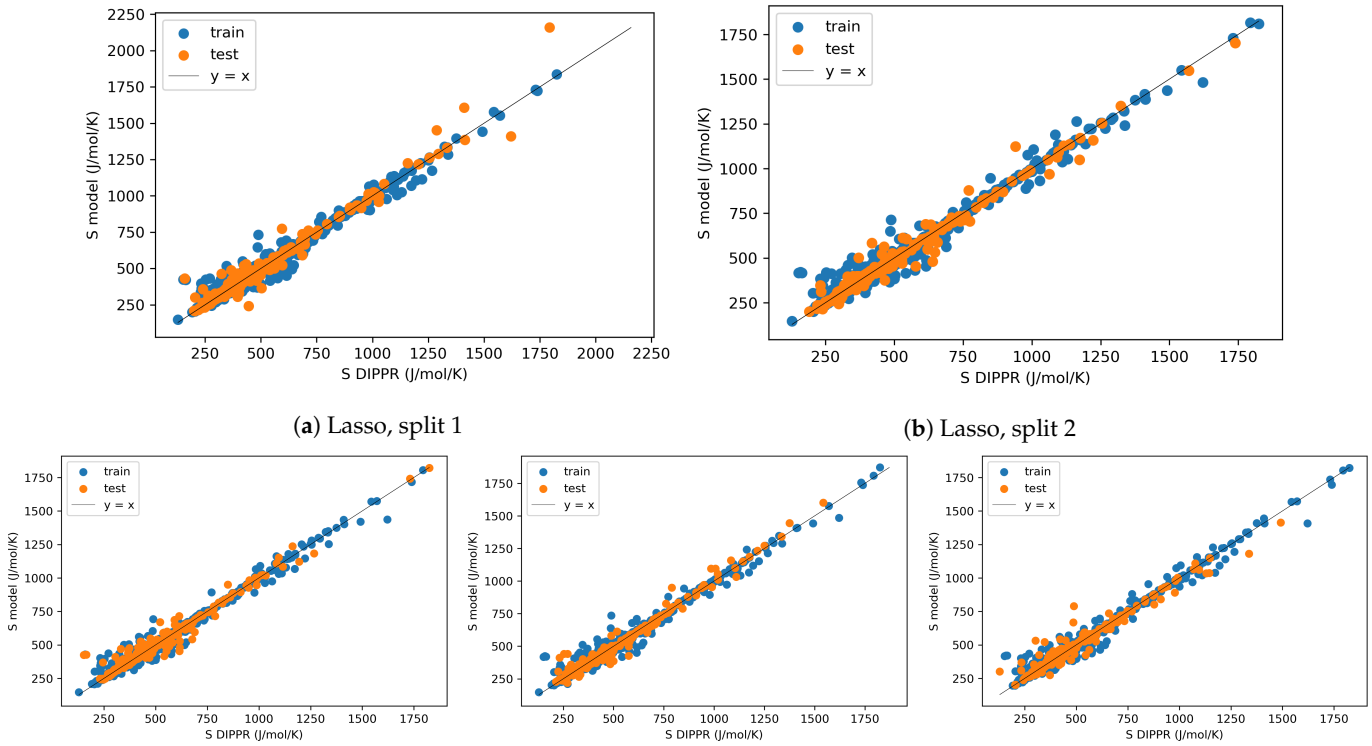


(**s**) MLP, split 4



(**t**) MLP, split 5

Figure 16. Parity plots of the selected ML models after HP optimization, for different splits, for the enthalpy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *yes*).



(**a**) Lasso, split 1

(**b**) Lasso, split 2

(**c**) Lasso, split 3

(**d**) Lasso, split 4

(**e**) Lasso, split 5

Figure 17. Parity plots of the selected ML models after HP optimization, for different splits, for the entropy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *yes*).

4. Benchmark

In this final part, the developed ML-QSPR procedure is benchmarked against other published works for the prediction of the enthalpy and the entropy. To ensure a fair comparison, the developed procedure (from data preprocessing to model construction) was applied to the same data sets as in the considered published works. The data preprocessing was composed of the elimination of the Desc-MVs by column (to ensure the use of the exactly same molecules but potentially leading to duplicated rows), the elimination of the descriptors with variance below 0.0001 and the elimination of correlated descriptors with a threshold of 0.98. As for the scaling method, a standard scaler was chosen. GA was then used to identify the 100 most important descriptors (cf. Supplementary Materials for the detailed list). Finally, a Lasso model was trained and validated via the nested CV scheme with $k = k'$. The value of k was chosen to have the same ratio between training (external) and test data as in the published works. Note, that some of them also used similar nested CV schemes.

The results of this benchmark study are presented in Table 16. It is interesting to observe that the performances are similar between this work and all the other published works, except the one of Dobbelaere et al. with the lignin QM data set for predicting the enthalpy [56]. Keeping in mind the significant reduction in the number of considered descriptors, it is noteworthy to observe that this work provides extremely comparable and, in some cases, improved performances than the established state-of-the-art in the domain. Besides these numerical comparisons, an added value of this work is also the meticulous break-down of the different steps and choices along the development procedure. The similar performances also evidence that there is no unique approach, in particular, there is no consensus on how to best represent molecular structures [63]. Each type of molecular representation displays its own advantages and drawbacks and the choice of a particular representation will depend on the requirements of each problem.

Table 16. [a] Hydrocarbons, oxygenated, nitrogenated, chlorinated, fluorinated, brominated, iodinated, phosphorus containing, sulfonated, silicon containing, multifunctional. [b] GroupGAT (group-contribution-based graph attention). [c] Probabilistic vector learned from interatomic distances, bond angles, and dihedral angles histograms with GMM (Gaussian Mixture Model). N/A: not available.

Property	Reference	Data Source	Type of Molecules	Nb of Molecules	Molecular Representation	ML Model	R^2 Test	*MAE* Test	*RMSE* Test	*k*
H (kJ/mol)	[109]	DIPPR *exp.*	Diverse [a]	741	GNN [b]	GNN [b]	0.99	18.6	30.5	10
	This work				100 descriptors	Lasso	0.99	12.4	21.6	
	[51]	Literature *exp., ab initio*	Noncyclic hydrocarbons	310	261 descriptors	SVR	0.995	5.703	N/A	10
	This work				100 descriptors	Lasso	0.998	4.426	6.520	
	[52]	Literature *exp.*	Cyclic hydrocarbons	192	47 descriptors	SVR	0.986	9.71	N/A	10
	[56]				GauL HDAD [c]	ANN	N/A	9.6	12.9	
	This work				100 descriptors	Lasso	0.985	10.01	14.51	
	[56]	Lignin QM *ab initio*	(Poly)cyclic hydrocarbons and oxygenates	3926	GauL HDAD [c]	ANN	N/A	9.34	15.89	10
	This work				100 descriptors	Lasso	0.98	21.48	30.81	
	[110]	SPEED *exp.*	Diverse	1059	240 groups	GP	0.987	N/A	42.74	20
	This work				100 descriptors	Lasso	0.976	11.30	28.20	
S (J/mol/K)	[56]	Lignin QM *ab initio*	(Poly)cyclic hydrocarbons and oxygenates	3926	GauL HDAD [c]	ANN	N/A	3.86	5.32	10
	This work				100 descriptors	Lasso	0.99	5.57	7.43	
	[53]	Literature *exp., theo.*	Hydrocarbons	310	252 descriptors	SVR	0.99	6.3	9	10
	This work				100 descriptors	Lasso	0.98	8.3	10.8	
	[111]	DIPPR *exp.*	Organic	511	GNN	GNN	0.99	5.3	N/A	10
	This work				100 descriptors	Lasso	0.99	6.1	9.4	

5. Conclusions and Perspectives

In this work, two ML-QSPR models were developed to predict the enthalpy of formation and the entropy of molecules from their structural and physico-chemical characteristics, represented by descriptors. The essence of this study lies in the adopted multi-angle perspective which provides a better overview of the possible methods at each step of the ML-QSPR procedure (i.e., data preprocessing, dimensionality reduction and model construction) and an understanding of the effects related to a given choice or method on the model performance, interpretability and applicability domain. Another characteristic of this study is the complexity of the data set which comprises a high diversity of molecules (to increase the applicability domain) and a high-dimensional descriptor-based molecular representation (to increase the chances of capturing the relevant features affecting the thermodynamic properties, in absence of knowledge). This was successfully addressed through customized data preprocessing techniques and genetic algorithms. The former improves the data quality while limiting the loss of information which, therefore, avoids applicability domain reduction and loss in the differentiation of the molecules. The latter allows for an automatic (i.e., in the absence of domain expert knowledge) identification of the most important descriptors to improve model interpretability, and the identified descriptors were found to be consistent with the physics. Finally, with the obtained data set, the best prediction performances were reached with a Lasso linear model (*MAE* test = 25.2 kJ/mol

for the enthalpy and 17.9 J/mol/K for the entropy), interpretable via the linear model coefficients. The overall developed procedure was also tested on various enthalpy and entropy related data sets from the literature to check its applicability to other problems and similar performances as those in the literature were obtained. This highlights that different methods and molecular representations, not necessarily the most complex ones, can lead to good performances. In any case, the retained methods and choices in any QSPR/QSAR model are problem specific, meaning that a different problem (i.e., with different requirements in terms of model precision, interpretability or computation time, and with different data characteristics) would have led to another set of choices and methods. Even if the latter can not be clearly defined for each specific case, the multi-angle approach demonstrated here is expected to provide a better overview and understanding of the methods and choices that could be applied in similar high-dimensional QSPR/QSAR problems.

However, the procedure is obviously improvable in several aspects. First of all, one of the OECD principles for the validation of QSAR/QSPR models was not addressed, namely the applicability domain of the models. This is crucial as the final goal of a QSPR/QSAR model is to be applied to new molecules and it is known that a ML model is not extrapolable. The applicability domain corresponds to the response and chemical structure space within which the model can make predictions with a given reliability. In this work, only a wide diversity of molecules and a customized pretreatment process were considered to "maximize" the applicability of the model to a large range of molecular structures. The next article of this series will be exclusively dedicated to the applicability domain definition of the developed models [89]. In particular, methods more adapted to high-dimensional data (as is the case in this problem) will be investigated at different steps of the ML-QSPR procedure to define the applicability domain (correspondingly, to detect the outliers). At the same time, this will help to address the overfitting phenomena which were observed for the developed models.

Concerning the data collection step, several ways of improvement can be envisioned. The conversion procedure from SMILES to descriptors requires further analysis. For example, it is not well understood how precise or reliable are the ETKDG method and AlvaDesc descriptor calculation with bigger, more complex or exotic molecules. Also, the uncertainties in descriptor values are unknown. Besides, the SMILES notation seems not adapted to differentiate some molecules, resulting in identical descriptors. Another improvement point concerns the diversity (i.e., in terms of structure and property) of the considered molecules and their unequal distribution. This questions the eventual influence that the most represented molecules could have on the developed models and the feasibility of building generic models applicable to all molecules. This diversity was particularly problematic, as some descriptors contained missing values for some types of molecules. This resulted in a loss of information during data preprocessing (elimination of molecules and descriptors with missing values), overfitting as well as high variability in the identified descriptors and model performances depending on the data split. A possible solution would be to create different models, one for each "category" of molecules. However, the best way to categorize the molecules needs to be investigated (e.g., by identifying clusters of molecules or based on chemical families) and it is likely that some categories will contain very low amounts of data. Regarding the considered chemical families in this study, some are generally removed in similar studies in the literature, such as inorganic compounds. The consideration or the separation (from the rest of the data set) of these molecules needs further analysis. In general, inorganic and organometallic compounds, counterions, salts and mixtures are removed during data collection or pretreatment, as they can not be handled by conventional cheminformatics techniques [28].

Above all, the molecular representation requires intensive study. Indeed, this work highlights several limitations of descriptors, namely their high-dimensional character, the lack of their understanding (for non-experts) or their unavailability for some molecules. Molecular representation is a particularly active area of research and an example of a recent and interesting method is graph-based representations (a.k.a. graph neural networks). The

latter internally combines feature extraction, which learns the important features from an initial molecular graph representation, and model construction, to relate the features to the target property. The main advantage of this type of representation lies in its capacity to automatically learn the molecular representation adapted for a specific problem, avoiding the laborious task of descriptor selection prior to model construction. Additionally, a QSPR model is based on the similarity principle (i.e., similar structures have similar properties) and on the assumption that the adopted molecular representation effectively contains all the information necessary to explain the studied property. While the first assumption is difficult to verify, the second could be addressed with other molecular representations. For all these reasons, graph-based representations could be envisioned. Besides, as each molecular representation contains different structural features, potentially interesting for predicting a given property, a combination of various representations (e.g., descriptors, fingerprints, graphs) could be investigated as well.

More generally, despite the provided multi-angle approach, the list of the presented methods is not exhaustive and some methods can be tested or further optimized. Some examples are listed below:

- identification of non-linearly correlated descriptors during data preprocessing;
- optimization of the HPs in the methods for dimensionality reduction (e.g., model and HPs in wrapper methods, HPs in embedded methods, number of selected descriptors);
- combination of different dimensionality reduction methods (sequentially; or in parallel followed by the union or intersection of the identified descriptors);
- other HP optimization techniques, less time consuming and more efficient than GridSearchCV;
- parallelization or use of computer clusters to reduce computation time;
- better consideration by the model of the uncertainties in property values;
- sensitivity analysis to determine the contribution of the descriptors on the predicted properties;
- comparison with GC or QC methods.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/pr11123325/s1, S1 (pdf): S1-Details on the methods and additional results; S2 (excel): S2-Data and ML predictions.

Author Contributions: C.T.: literature review, conceptualization, methodology, data curation and modeling, writing (original draft preparation, review and editing). Y.T.: data curation and modeling, development of the graph theory based method for the elimination of correlations between descriptors. D.M.: supervision, methodology, writing (review and editing). S.L. and O.H.: data provision, molecular and thermodynamic analyses, writing (review and editing). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by MESRI (Ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation), and by the Institute Carnot ICEEL (Grant: "Recyclage de Pneus par Intelligence Artificielle - RePnIA"), France.

Data Availability Statement: The authors do not have the permission to share the data from DIPPR, only some information on the descriptors and the predictions as well as additional results are available in the Supplementary Materials File S1 (pdf) and File S2 (excel).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AB	Adaptive boosting
CV	Cross-validation
Desc-MVs	Missing descriptor values
DIPPR	Design institute for physical properties
DT	Decision tree
ET	Extra trees
ETKDG	Experimental torsion distance geometry with additional basic knowledge terms
H	Enthalpy for ideal gas at 298.15 K and 1 bar
GA	Genetic algorithm
GB	Gradient boosting
GC	Group contribution
GNN	Graph neural network
GP	Gaussian processes
HP	Hyperparameter
kNN	k-nearest neighbors
Lasso	Least absolute shrinkage and selection operator
LDA	Linear discriminant analysis
LR	Linear regression (ordinary least squares)
MAE	Mean absolute error
MI	Mutual information
ML	Machine learning
MLP	Multilayer perceptron
OECD	Organisation for economic co-operation and development
PCs	Principal components
PCA	Principal component analysis
QC	Quantum chemistry
QSAR	Quantitative structure-activity relationship
QSPR	Quantitative structure-property relationship
R^2	Coefficient of determination
RF	Random forest
RMSE	Root mean square error
S	Absolute entropy of ideal gas at 298.15 K and 1 bar
SFS	Sequential forward selection
SMILES	Simplified molecular input line entry specification
SVR	Support vector regression
SVR lin	Linear support vector regression

References

1. Rao, H.; Zhu, Z.; Le, Z.; Xu, Z. QSPR models for the critical temperature and pressure of cycloalkanes. *Chem. Phys. Lett.* **2022**, *808*, 140088. [CrossRef]
2. Roubehie Fissa, M.; Lahiouel, Y.; Khaouane, L.; Hanini, S. QSPR estimation models of normal boiling point and relative liquid density of pure hydrocarbons using MLR and MLP-ANN methods. *J. Mol. Graph. Model.* **2019**, *87*, 109–120. [CrossRef] [PubMed]
3. Bloxham, J.; Hill, D.; Giles, N.F.; Knotts, T.A.; Wilding, W.V. New QSPRs for Liquid Heat Capacity. *Mol. Inform.* **2022**, *41*, 1–7. [CrossRef] [PubMed]
4. Yu, X.; Acree, W.E. QSPR-based model extrapolation prediction of enthalpy of solvation. *J. Mol. Liq.* **2023**, *376*, 121455. [CrossRef]
5. Jia, Q.; Yan, X.; Lan, T.; Yan, F.; Wang, Q. Norm indexes for predicting enthalpy of vaporization of organic compounds at the boiling point. *J. Mol. Liq.* **2019**, *282*, 484–488. [CrossRef]
6. Yan, X.; Lan, T.; Jia, Q.; Yan, F.; Wang, Q. A norm indexes-based QSPR model for predicting the standard vaporization enthalpy and formation enthalpy of organic compounds. *Fluid Phase Equilibria* **2020**, *507*, 112437. [CrossRef]
7. Mauri, A.; Bertola, M. Alvascience: A New Software Suite for the QSAR Workflow Applied to the Blood–Brain Barrier Permeability. *Int. J. Mol. Sci.* **2022**, *23*, 12882. [CrossRef] [PubMed]
8. Rasulev, B.; Casanola-Martin, G. QSAR/QSPR in Polymers. *Int. J. Quant.-Struct.-Prop. Relationships* **2020**, *5*, 80–88. [CrossRef]
9. Zhang, Y.; Xu, X. Machine learning glass transition temperature of polyacrylamides using quantum chemical descriptors. *Polym. Chem.* **2021**, *12*, 843–851. [CrossRef]

10. Schustik, S.A.; Cravero, F.; Ponzoni, I.; Díaz, M.F. Polymer informatics: Expert-in-the-loop in QSPR modeling of refractive index. *Comput. Mater. Sci.* **2021**, *194*, 110460. [CrossRef]
11. Li, R.; Herreros, J.M.; Tsolakis, A.; Yang, W. Machine learning-quantitative structure property relationship (ML-QSPR) method for fuel physicochemical properties prediction of multiple fuel types. *Fuel* **2021**, *304*, 121437. [CrossRef]
12. Sun, Y.; Chen, M.C.; Zhao, Y.; Zhu, Z.; Xing, H.; Zhang, P.; Zhang, X.; Ding, Y. Machine learning assisted QSPR model for prediction of ionic liquid's refractive index and viscosity: The effect of representations of ionic liquid and ensemble model development. *J. Mol. Liq.* **2021**, *333*, 115970. [CrossRef]
13. Paduszyński, K.; Kłębowski, K.; Królikowska, M. Predicting melting point of ionic liquids using QSPR approach: Literature review and new models. *J. Mol. Liq.* **2021**, *344*, 117631. [CrossRef]
14. Sepehri, B. A review on created QSPR models for predicting ionic liquids properties and their reliability from chemometric point of view. *J. Mol. Liq.* **2020**, *297*, 112013. [CrossRef]
15. Yan, F.; Shi, Y.; Wang, Y.; Jia, Q.; Wang, Q.; Xia, S. QSPR models for the properties of ionic liquids at variable temperatures based on norm descriptors. *Chem. Eng. Sci.* **2020**, *217*, 115540. [CrossRef]
16. Zhu, T.; Chen, Y.; Tao, C. Multiple machine learning algorithms assisted QSPR models for aqueous solubility: Comprehensive assessment with CRITIC-TOPSIS. *Sci. Total. Environ.* **2023**, *857*, 159448. [CrossRef] [PubMed]
17. Duchowicz, P.R. QSPR studies on water solubility, octanol-water partition coefficient and vapour pressure of pesticides. *SAR QSAR Environ. Res.* **2020**, *31*, 135–148. [CrossRef] [PubMed]
18. Euldji, I.; Si-Moussa, C.; Hamadache, M.; Benkortbi, O. QSPR Modelling of the Solubility of Drug and Drug-like Compounds in Supercritical Carbon Dioxide. *Mol. Inform.* **2022**, *41*, 1–16. [CrossRef]
19. Meftahi, N.; Walker, M.L.; Smith, B.J. Predicting aqueous solubility by QSPR modeling. *J. Mol. Graph. Model.* **2021**, *106*, 107901. [CrossRef]
20. Raevsky, O.A.; Grigorev, V.Y.; Polianczyk, D.E.; Raevskaja, O.E.; Dearden, J.C. Aqueous Drug Solubility: What Do We Measure, Calculate and QSPR Predict? *Mini-Rev. Med. Chem.* **2019**, *19*, 362–372. [CrossRef]
21. Chinta, S.; Rengaswamy, R. Machine Learning Derived Quantitative Structure Property Relationship (QSPR) to Predict Drug Solubility in Binary Solvent Systems. *Ind. Eng. Chem. Res.* **2019**, *58*, 3082–3092. [CrossRef]
22. Chaudhari, P.; Ade, N.; Pérez, L.M.; Kolis, S.; Mashuga, C.V. Quantitative Structure-Property Relationship (QSPR) models for Minimum Ignition Energy (MIE) prediction of combustible dusts using machine learning. *Powder Technol.* **2020**, *372*, 227–234. [CrossRef]
23. Bouarab-Chibane, L.; Forquet, V.; Lantéri, P.; Clément, Y.; Léonard-Akkari, L.; Oulahal, N.; Degraeve, P.; Bordes, C. Antibacterial properties of polyphenols: Characterization and QSAR (Quantitative structure-activity relationship) models. *Front. Microbiol.* **2019**, *10*, 829. [CrossRef] [PubMed]
24. Kirmani, S.A.K.; Ali, P.; Azam, F. Topological indices and QSPR/QSAR analysis of some antiviral drugs being investigated for the treatment of COVID-19 patients. *Int. J. Quantum Chem.* **2021**, *121*, 1–22. [CrossRef] [PubMed]
25. Cherkasov, A.; Muratov, E.N.; Fourches, D.; Varnek, A.; Baskin, I.I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y.C.; Todeschini, R.; et al. QSAR modeling: Where have you been? Where are you going to? *J. Med. Chem.* **2014**, *57*, 4977–5010. [CrossRef] [PubMed]
26. Yousefinejad, S.; Hemmateenejad, B. Chemometrics tools in QSAR/QSPR studies: A historical perspective. *Chemom. Intell. Lab. Syst.* **2015**, *149*, 177–204. [CrossRef]
27. Liu, P.; Long, W. Current mathematical methods used in QSAR/QSPR studies. *Int. J. Mol. Sci.* **2009**, *10*, 1978–1998. [CrossRef]
28. Tropsha, A. Best practices for QSAR model development, validation, and exploitation. *Mol. Inform.* **2010**, *29*, 476–488. [CrossRef]
29. Gramatica, P. *A Short History of QSAR Evolution*; Insubria University: Varese, Italy, 2011.
30. He, C.; Zhang, C.; Bian, T.; Jiao, K.; Su, W.; Wu, K.J.; Su, A. A Review on Artificial Intelligence Enabled Design, Synthesis, and Process Optimization of Chemical Products for Industry 4.0. *Processes* **2023**, *11*, 330. [CrossRef]
31. Kuntz, D.; Wilson, A.K. Machine learning, artificial intelligence, and chemistry: How smart algorithms are reshaping simulation and the laboratory. *Pure Appl. Chem.* **2022**, *94*, 1019–1054. [CrossRef]
32. Toropov, A.A. QSPR/QSAR: State-of-Art, Weirdness, the Future. *Molecules* **2020**, *25*, 1292. [CrossRef] [PubMed]
33. Dearden, J.C.; Cronin, M.T.; Kaiser, K.L. How not to develop a quantitative structure-activity or structure-property relationship (QSAR/QSPR). *SAR QSAR Environ. Res.* **2009**, *20*, 241–266. [CrossRef] [PubMed]
34. OECD. Guidance Document on the Validation of (Quantitative) Structure-Activity Relationship [(Q)SAR] Models; OECD: Paris, France, 2007.
35. Dral, P.O. Quantum Chemistry in the Age of Machine Learning. *J. Phys. Chem. Lett.* **2020**, *11*, 2336–2347. [CrossRef] [PubMed]
36. Narayanan, B.; Redfern, P.C.; Assary, R.S.; Curtiss, L.A. Accurate quantum chemical energies for 133000 organic molecules. *Chem. Sci.* **2019**, *10*, 7449–7455. [CrossRef] [PubMed]
37. Zhao, Q.; Savoie, B.M. Self-Consistent Component Increment Theory for Predicting Enthalpy of Formation. *J. Chem. Inf. Model.* **2020**, *60*, 2199–2207. [CrossRef] [PubMed]
38. Grambow, C.A.; Li, Y.P.; Green, W.H. Accurate Thermochemistry with Small Data Sets: A Bond Additivity Correction and Transfer Learning Approach. *J. Phys. Chem. A* **2019**, *123*, 5826–5835. [CrossRef] [PubMed]

39. Li, Q.; Wittreich, G.; Wang, Y.; Bhattacharjee, H.; Gupta, U.; Vlachos, D.G. Accurate Thermochemistry of Complex Lignin Structures via Density Functional Theory, Group Additivity, and Machine Learning. *ACS Sustain. Chem. Eng.* **2021**, *9*, 3043–3049. [CrossRef]
40. Gu, G.H.; Plechac, P.; Vlachos, D.G. Thermochemistry of gas-phase and surface species via LASSO-assisted subgraph selection. *React. Chem. Eng.* **2018**, *3*, 454–466. [CrossRef]
41. Gertig, C.; Leonhard, K.; Bardow, A. Computer-aided molecular and processes design based on quantum chemistry: Current status and future prospects. *Curr. Opin. Chem. Eng.* **2020**, *27*, 89–97. [CrossRef]
42. Cao, Y.; Romero, J.; Olson, J.P.; Degroote, M.; Johnson, P.D.; Kieferová, M.; Kivlichan, I.D.; Menke, T.; Peropadre, B.; Sawaya, N.P.; et al. Quantum Chemistry in the Age of Quantum Computing. *Chem. Rev.* **2019**, *119*, 10856–10915. [CrossRef]
43. Constantinou, L.; Gani, R. New group contribution method for estimating properties of pure compounds. *AIChE J.* **1994**, *40*, 1697–1710. [CrossRef]
44. Marrero, J.; Gani, R. Group-contribution based estimation of pure component properties. *Fluid Phase Equilibria* **2001**, *183–184*, 183–208. [CrossRef]
45. Trinh, C.; Meimaroglou, D.; Hoppe, S. Machine learning in chemical product engineering: The state of the art and a guide for newcomers. *Processes* **2021**, *9*, 1456. [CrossRef]
46. RDKit: Open-Source Cheminformatics. Available online: https://www.rdkit.org/docs/index.html (accessed on 1 June 2023).
47. Mauri, A. alvaDesc: A tool to calculate and analyze molecular descriptors and fingerprints. In *Ecotoxicological QSARs: Methods in Pharmacology and Toxicology*; Humana: New York, NY, USA, 2020; pp. 801–820. [CrossRef]
48. Yap, C.W. PaDEL-Descriptor: An Open Source Software to Calculate Molecular Descriptors and Fingerprints. *J. Comput. Chem.* **2010**, *32*, 174–182. [CrossRef]
49. Steinbeck, C.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E. The Chemistry Development Kit (CDK): An open-source Java library for chemo- and bioinformatics. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 493–500. [CrossRef] [PubMed]
50. Moriwaki, H.; Tian, Y.S.; Kawashita, N.; Takagi, T. Mordred: A molecular descriptor calculator. *J. Cheminformatics* **2018**, *10*, 1–14. [CrossRef] [PubMed]
51. Yalamanchi, K.K.; Van Oudenhoven, V.C.; Tutino, F.; Monge-Palacios, M.; Alshehri, A.; Gao, X.; Sarathy, S.M. Machine Learning to Predict Standard Enthalpy of Formation of Hydrocarbons. *J. Phys. Chem. A* **2019**, *123*, 8305–8313. [CrossRef]
52. Yalamanchi, K.K.; Monge-Palacios, M.; Van Oudenhoven, V.C.; Gao, X.; Sarathy, S.M. Data Science Approach to Estimate Enthalpy of Formation of Cyclic Hydrocarbons. *J. Phys. Chem. A* **2020**, *124*, 6270–6276. [CrossRef]
53. Aldosari, M.N.; Yalamanchi, K.K.; Gao, X.; Sarathy, S.M. Predicting entropy and heat capacity of hydrocarbons using machine learning. *Energy AI* **2021**, *4*, 100054. [CrossRef]
54. Sheibani, N. Heat of Formation Assessment of Organic Azido Compounds Used as Green Energetic Plasticizers by QSPR Approaches. *Propellants Explos. Pyrotech.* **2019**, *44*, 1254–1262. [CrossRef]
55. Joudaki, D.; Shafiei, F. QSPR Models for the Prediction of Some Thermodynamic Properties of Cycloalkanes Using GA-MLR Method. *Curr. Comput. Aided Drug Des.* **2020**, *16*, 571–582. [CrossRef] [PubMed]
56. Dobbelaere, M.R.; Plehiers, P.P.; Van de Vijver, R.; Stevens, C.V.; Van Geem, K.M. Learning Molecular Representations for Thermochemistry Prediction of Cyclic Hydrocarbons and Oxygenates. *J. Phys. Chem. A* **2021**, *125*, 5166–5179. [CrossRef] [PubMed]
57. Wan, Z. Quantitative structure-property relationship of standard enthalpies of nitrogen oxides based on a MSR and LS-SVR algorithm predictions. *J. Mol. Struct.* **2020**, *1221*, 128867. [CrossRef]
58. DIPPR's Project 801 Database. Available online: https://www.aiche.org/dippr. (accessed on 1 June 2023).
59. Bloxham, J.C.; Redd, M.E.; Giles, N.F.; Knotts, T.A.; Wilding, W.V. Proper Use of the DIPPR 801 Database for Creation of Models, Methods, and Processes. *J. Chem. Eng. Data* **2020**, *66*, 3–10. [CrossRef]
60. Wigh, D.S.; Goodman, J.M.; Lapkin, A.A. A review of molecular representation in the age of machine learning. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2022**, *12*, 1–19. [CrossRef]
61. Wu, X.; Wang, H.; Gong, Y.; Fan, D.; Ding, P.; Li, Q.; Qian, Q. Graph neural networks for molecular and materials representation. *J. Mater. Inform.* **2023**, *3*, 12. [CrossRef]
62. Wieder, O.; Kohlbacher, S.; Kuenemann, M.; Garon, A.; Ducrot, P.; Seidel, T.; Langer, T. A compact review of molecular property prediction with graph neural networks. *Drug Discov. Today Technol.* **2020**, *37*, 1–12. [CrossRef] [PubMed]
63. Jiang, D.; Wu, Z.; Hsieh, C.Y.; Chen, G.; Liao, B.; Wang, Z.; Shen, C.; Cao, D.; Wu, J.; Hou, T. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *J. Cheminformatics* **2021**, *13*, 1–23. [CrossRef]
64. Van Tilborg, D.; Alenicheva, A.; Grisoni, F. Exposing the Limitations of Molecular Machine Learning with Activity Cliffs. *J. Chem. Inf. Model.* **2022**, *62*, 5938–5951. [CrossRef]
65. Orosz, Á.; Héberger, K.; Rácz, A. Comparison of Descriptor- and Fingerprint Sets in Machine Learning Models for ADME-Tox Targets. *Front. Chem.* **2022**, *10*, 1–15. [CrossRef]
66. Baptista, D.; Correia, J.; Pereira, B.; Rocha, M. Evaluating molecular representations in machine learning models for drug response prediction and interpretability. *J. Integr. Bioinform.* **2022**, *19*, 1–13. [CrossRef] [PubMed]
67. Riniker, S.; Landrum, G.A. Better Informed Distance Geometry: Using What We Know to Improve Conformation Generation. *J. Chem. Inf. Model.* **2015**, *55*, 2562–2574. [CrossRef] [PubMed]

68. Hawkins, P.C. Conformation Generation: The State of the Art. *J. Chem. Inf. Model.* **2017**, *57*, 1747–1756. [CrossRef] [PubMed]
69. Fourches, D.; Muratov, E.; Tropsha, A. Trust, but verify: On the importance of chemical structure curation in cheminformatics and QSAR modeling research. *J. Chem. Inf. Model.* **2010**, *50*, 1189–1204. [CrossRef] [PubMed]
70. Vabalas, A.; Gowen, E.; Poliakoff, E.; Casson, A.J. Machine learning algorithm validation with a limited sample size. *PLoS ONE* **2019**, *14*, e0224365. [CrossRef] [PubMed]
71. Wold, S. Principal Component Analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]
72. Bro, R.; Smilde, A.K. Principal component analysis. *Anal. Methods* **2014**, *6*, 2812–2831. [CrossRef]
73. Izenman, A.J. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*; Springer: Berlin/Heidelberg, Germany, 2008.
74. Dor, B.; Koenigstein, N.; Giryes, R. Autoencoders. *arXiv* **2020**, arXiv:2003.05991.
75. Doersch, C. Tutorial on Variational Autoencoders. *arXiv* **2016**, arXiv:1606.05908;
76. Saeys, Y.; Inza, I.; Larrañaga, P. A review of feature selection techniques in bioinformatics. *Bioinformatics* **2007**, *23*, 2507–2517. [CrossRef]
77. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. [CrossRef]
78. Bolón-Canedo, V.; Sánchez-Maroño, N.; Alonso-Betanzos, A. A review of feature selection methods on synthetic data. *Knowl. Inf. Syst.* **2013**, *34*, 483–519. [CrossRef]
79. Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R.P.; Tang, J.; Liu, H. Feature selection: A data perspective. *ACM Comput. Surv.* **2017**, *50*, 94. [CrossRef]
80. Kumar, V. Feature Selection: A literature Review. *Smart Comput. Rev.* **2014**, *4*, 211–229. [CrossRef]
81. Haury, A.C.; Gestraud, P.; Vert, J.P. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PLoS ONE* **2011**, *6*, e28210. [CrossRef] [PubMed]
82. Hira, Z.M.; Gillies, D.F. A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data. *Hindawi Publ. Corp. Adv. Bioinform.* **2015**, *2015*, 198363. [CrossRef] [PubMed]
83. Chen, C.W.; Tsai, Y.H.; Chang, F.R.; Lin, W.C. Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results. *Expert Syst.* **2020**, *37*, 1–10. [CrossRef]
84. Shahlaei, M. Descriptor selection methods in quantitative structure-activity relationship studies: A review study. *Chem. Rev.* **2013**, *113*, 8093–8103. [CrossRef]
85. Bommert, A.; Sun, X.; Bischl, B.; Rahnenführer, J.; Lang, M. Benchmark for filter methods for feature selection in high-dimensional classification data. *Comput. Stat. Data Anal.* **2020**, *143*, 106839. [CrossRef]
86. Mangal, A.; Holm, E.A. A Comparative Study of Feature Selection Methods for Stress Hotspot Classification in Materials. *Integr. Mater. Manuf. Innov.* **2018**, *7*, 87–95. [CrossRef]
87. Eklund, M.; Norinder, U.; Boyer, S.; Carlsson, L. Choosing feature selection and learning algorithms in QSAR. *J. Chem. Inf. Model.* **2014**, *54*, 837–843. [CrossRef] [PubMed]
88. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
89. Trinh, C.; Lasala, S.; Herbinet, O.; Meimaroglou, D. On the Development of Descriptor-Based Machine Learning Models for Thermodynamic Properties. Part 2—Applicability Domain and Outliers. *Algorithms under review*.
90. Cawley, G.C.; Talbot, N.L. On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.* **2010**, *11*, 2079–2107.
91. Krstajic, D.; Buturovic, L.J.; Leahy, D.E.; Thomas, S. Cross-validation pitfalls when selecting and assessing regression and classification models. *J. Cheminformatics* **2014**, *6*, 1–15. [CrossRef] [PubMed]
92. Anguita, D.; Ghelardoni, L.; Ghio, A.; Oneto, L.; Ridella, S. The 'K' in K-fold cross validation. In Proceedings of the ESANN 2012 Proceedings, 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, 25–27 April 2012; pp. 441–446.
93. Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In Proceedings of the International Joint Conference of Artificial Intelligence, Montreal, QC, Canada, 20–25 August 1995.
94. Gramatica, P.; Sangion, A. A Historical Excursus on the Statistical Validation Parameters for QSAR Models: A Clarification Concerning Metrics and Terminology. *J. Chem. Inf. Model.* **2016**, *56*, 1127–1131. [CrossRef] [PubMed]
95. Chirico, N.; Gramatica, P. Real external predictivity of QSAR models: How to evaluate It? Comparison of different validation criteria and proposal of using the concordance correlation coefficient. *J. Chem. Inf. Model.* **2011**, *51*, 2320–2335. [CrossRef] [PubMed]
96. Yang, L.; Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, *415*, 295–316. [CrossRef]
97. Hastie, T.; Friedman, J.; Tisbshirani, R. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer: Berlin/Heidelberg, Germany, 2017.
98. Vapnik, V.N. *The Nature of Statistical Learning*; Springer: New York, NY, USA, 1995.
99. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
100. Verleysen, M.; François, D. The Curse of Dimensionality in Data Mining and Time Series Prediction. In Proceedings of the 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Barcelona, Spain, 8–10 June 2005.

101. Aggarwal, C.C.; Yu, P.S. Outlier detection for high dimensional data. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Santa Barbara, CA, USA, 21–24 May 2001; pp. 37–46. [CrossRef]
102. Pfingstl, S.; Zimmermann, M. On integrating prior knowledge into Gaussian processes for prognostic health monitoring. *Mech. Syst. Signal Process.* **2022**, *171*, 108917. [CrossRef]
103. Hallemans, N.; Pintelon, R.; Peumans, D.; Lataire, J. Improved frequency response function estimation by Gaussian process regression with prior knowledge. *IFAC-PapersOnLine* **2021**, *54*, 559–564. [CrossRef]
104. Long, D.; Wang, Z.; Krishnapriyan, A.; Kirby, R.; Zhe, S.; Mahoney, M. AutoIP: A United Framework to Integrate Physics into Gaussian Processes. arXiv **2022**, arXiv:2202.12316.
105. Han, K.; Jamal, A.; Grambow, C.A.; Buras, Z.J.; Green, W.H. An Extended Group Additivity Method for Polycyclic Thermochemistry Estimation. *Int. J. Chem. Kinet.* **2018**, *50*, 294–303. [CrossRef]
106. Zhao, Q.; Iovanac, N.C.; Savoie, B.M. Transferable Ring Corrections for Predicting Enthalpy of Formation of Cyclic Compounds. *J. Chem. Inf. Model.* **2021**, *61*, 2798–2805. [CrossRef] [PubMed]
107. Li, Y.P.; Han, K.; Grambow, C.A.; Green, W.H. Self-Evolving Machine: A Continuously Improving Model for Molecular Thermochemistry. *J. Phys. Chem. A* **2019**, *123*, 2142–2152. [CrossRef] [PubMed]
108. Lay, T.H.; Yamada, T.; Tsai, P.L.; Bozzelli, J.W. Thermodynamic parameters and group additivity ring corrections for three- to six-membered oxygen heterocyclic hydrocarbons. *J. Phys. Chem. A* **1997**, *101*, 2471–2477. [CrossRef]
109. Aouichaoui, A.R.; Fan, F.; Mansouri, S.S.; Abildskov, J.; Sin, G. Combining Group-Contribution Concept and Graph Neural Networks Toward Interpretable Molecular Property Models. *J. Chem. Inf. Model.* **2023**, *63*, 725–744. [CrossRef]
110. Alshehri, A.S.; Tula, A.K.; You, F.; Gani, R. Next generation pure component property estimation models: With and without machine learning techniques. *AIChE J.* **2021**, *68*, e17469. [CrossRef]
111. Aouichaoui, A.R.; Fan, F.; Abildskov, J.; Sin, G. Application of interpretable group-embedded graph neural networks for pure compound properties. *Comput. Chem. Eng.* **2023**, *176*, 108291. [CrossRef]

2.4 Supplementary Materials

The present article is accompanied by two supplementary materials, one PDF file (Supplementary Material 1) and one Excel file (Supplementary Material 2). The latter will be provided during the submission to the journal *Algorithms* while the former is provided in this section and consists in the following parts:

1. **Comparison of the pairwise method and the graph-based method for the elimination of correlated descriptors**
2. **Genetic algorithm for dimensionality reduction**
3. **Additional results for the enthalpy**
   - 3.1 Data preprocessing
   - 3.2 Screening with final preprocessing options and wrapper-GA Lasso method for dimensionality reduction
4. **Results for the entropy**
   - 4.1 Preliminary screening with default preprocessing and without dimensionality reduction
   - 4.2 Data preprocessing
   - 4.3 Dimensionality reduction
   - 4.4 Screening with final preprocessing options and wrapper-GA Lasso method for dimensionality reduction
   - 4.5 HP optimization
5. **Information on the AlvaDesc descriptor categories identified during dimensionality reduction**
6. **Additional information on the tested tools for descriptor calculation**

processes

MDPI

Article

S1-Details on the methods and additional results On the Development of Descriptor-Based Machine Learning Models for Thermodynamic Properties Part 1 - From Data Collection to Model Construction: Understanding of the Methods and their Effects

Cindy Trinh, Youssef Tbatou, Silvia Lasala, Olivier Herbinet and Dimitrios Meimaroglou *

Université de Lorraine, CNRS, LRGP, F-54001 Nancy, France; cindy.trinh.ct@outlook.com (C.T.); yousseftbatou3@gmail.com (Y.T.); silvia.lasala@univ-lorraine.fr (S.L.); olivier.herbinet@univ-lorraine.fr (O.H.)
* Correspondence: dimitrios.meimaroglou@univ-lorraine.fr

S1. Comparison of the pairwise method and the graph-based method for the elimination of correlated descriptors

In a pairwise treatment for the elimination of the correlated descriptors, these are listed in pairs presenting a correlation coefficient value that is superior to a predefined fixed threshold. Subsequently, only one of the two descriptors of each pair (e.g., the first one) is retained and the other one is discarded. However, in a high-dimensional data set, multiple correlations may exist that would render this approach insufficient. Consider, for example, the case where a descriptor B is correlated with two descriptors A and C, forming therefore the pairs A-B and B-C. According to the described procedure, the first descriptor of each pair, namely A and B, would be retained, but here A and B are still correlated. Additionally, it is not possible to keep only A, otherwise the information in C will be lost (C is not correlated to A and correlation is not transitive). However, having remaining correlations could be problematic as it increases computation cost and risk of overfitting regarding the high dimension of the data set. The best choice in this example would have been to keep A and C.

A new method was therefore developed in this work to better eliminate the correlated descriptors in a high dimensional space. It is based on graphs, where the nodes represent the descriptors while the edges link the descriptors that are correlated above a given threshold.

An example is shown here to compare both approaches in the elimination of the correlated descriptors. The example considers 8 descriptors (A to H) whose correlation matrix is presented in Table S1. The considered threshold is 0.75, meaning that the goal is to eliminate descriptors so that there are no correlations above 0.75 left among the remaining descriptors after the elimination procedure. The correlations above this threshold are shown in blue in Table S1, where only the upper triangular matrix without the diagonal is considered. The resulting list of pairwise correlations is described in Table S2 while Figure S1 shows the corresponding graph. Following the pairwise procedure, only the first descriptor of each pair is conserved and the final descriptors are: A, B, C, D and E. However, it is clearly visible that there are still existing correlations among these final descriptors. According to the graph-based method, the first step is to keep the descriptor having the highest number of correlations (B in this case), and eliminate the descriptors which are correlated to B (namely C, D and G). Then, the descriptors that were not analyzed yet are A, E, F and H. Again, the descriptor having the highest number of correlations is kept (H in this case), and the descriptors which are correlated to H are eliminated (namely A and E). Finally, F is also conserved. To resume, the final descriptors are B, H and F and they are not correlated.

Table S1. Example of a correlation matrix (correlations above 0.75 are shown in blue).

	A	B	C	D	E	F	G	H
A	1.00	**0.96**	0.42	0.68	0.28	0.44	0.71	**0.96**
B	0.96	1.00	**0.92**	**0.76**	0.05	0.38	**0.75**	0.34
C	0.42	0.92	1.00	0.74	0.10	**0.77**	0.28	0.59
D	0.68	0.76	0.74	1.00	**0.82**	**0.80**	0.68	0.22
E	0.28	0.05	0.10	0.82	1.00	0.19	0.66	**0.75**
F	0.44	0.38	0.77	0.80	0.19	1.00	0.16	0.26
G	0.71	0.75	0.28	0.68	0.66	0.16	1.00	0.51
H	0.96	0.34	0.59	0.22	0.75	0.26	0.51	1.00

Table S2. List of pairwise correlations above 0.75 based on the correlation matrix in Table S1.

Descriptor 1	Descriptor 2
A	B
A	H
B	C
B	D
B	G
C	F
D	E
D	F
E	H

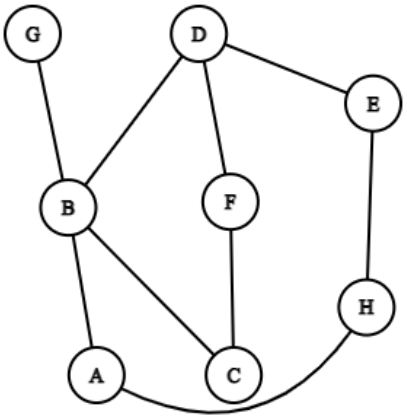


Figure S1. Graph based on the correlation matrix in Table S1.

S2. Genetic algorithm for dimensionality reduction

Genetic algorithms (GA) are a powerful tool employed to identify solutions to optimization problems by relying on a procedure inspired from natural selection through genetic operators (e.g., selection, crossover and mutation). Figure S2 represents a typical workflow of a GA procedure, which is the one used in this work to identify the 100 best descriptors during dimensionality reduction step. An initial population of candidate solutions (a.k.a. individuals) is firstly generated randomly. This population then evolves across several generations through genetic operators until an optimal solution is identified. These operations are described in the following.

Creation of the initial population. The initial population is composed of N_{ind} individuals. Each individual is a vector of size D_{init}, the initial number of dimensions or descriptors (i.e., at the beginning of the dimensionality reduction step). This vector (a.k.a. genome) contains D_{final} 1s which indicate the selection of the corresponding descriptors, D_{final} being the number of descriptors that need to be selected from the initial D_{init} descriptors. The rest of the vector is exclusively composed of 0s.

Evaluation of the individuals. The individuals of this initial population are subsequently evaluated via a fitness function. In this work, each individual (i.e. a subset of descriptors) was used to train a Lasso model with default HPs and the obtained MAE was used to evaluate the individual. The lowest MAE, the best individual.

Stopping criterion. Different stopping criteria can be considered in a GA such as a maximum number of generations N_{gen} (this work) or a MAE value below a given threshold for the best individual. If so, the optimal solution is found. Otherwise, a new population of N_{ind} individuals is generated from the previous one via genetic operators until the stopping criterion is satisfied.

New population: Storage of the two best individuals. To generate a new population, the two best individuals of the previous population (i.e., with the lowest MAE) are first conserved. ***Selection.*** To generate the other individuals of the new population, a selection step is then needed to select the parents from the previous population. In this work, each pair of parents was selected randomly among the 40% best individuals from the previous population to generate two new individuals or children, until reaching the required number of individuals N_{ind}. The genomes of each pair of parents are then submitted to a crossover, a mutation or a crossover+mutation procedure, with respective probabilities of p_c, p_m and p_{c+m}. ***Crossover.*** In the crossover procedure (also called recombination), the genomes of the two parents are recombined to generate the two new child genomes. Different recombination techniques exist but in this work, the following one was used: each child's genome initially contains D_{init} 0s, which are turning into 1s at the locations (randomly chosen) where at least one of the parents contains 1-value, until reaching D_{final} 1s. ***Mutation.*** As for the mutation operator, it consists in randomly modifying the genomes of the parents to introduce diversity in the new population and therefore limiting the risk to converge towards a local minima. In this study, a random mutation percentage between 1 and 20% is applied to the genome of each parent, with half 1-to-0 mutations and half 0-to-1 mutations to keep the number of 1s always equal to D_{final}. ***Crossover+mutation.*** Finally, in the crossover+mutation procedure, the genomes of the parents are subject to a crossover step, which resulting genomes are subsequently subject to mutation. ***Evaluation of the children.*** After each procedure (crossover, mutation or crossover+mutation), the two resulting child genomes are evaluated via the same fitness function as mentioned earlier. A child is included in the new population only if its MAE is better than the worst of the previous population. Otherwise, one of the 40% best individuals from the previous population is added to the new population instead. Once the new population is complete, the same workflow as previously is applied starting with the evaluation of the individuals.

In general, it is recommended to repeat the GA procedure (Figure S2) several times to reduce the risk to converge towards a local minima. This was done three times in this work during dimensionality reduction step, without showing significant MAE differences in the

identified solutions. However, for the benchmark, the procedure was implemented only once.

Also, note that the HPs of the GA were not optimized in this work but this could be envisioned as a future step. However, the values that were used are summarized in Table S3.

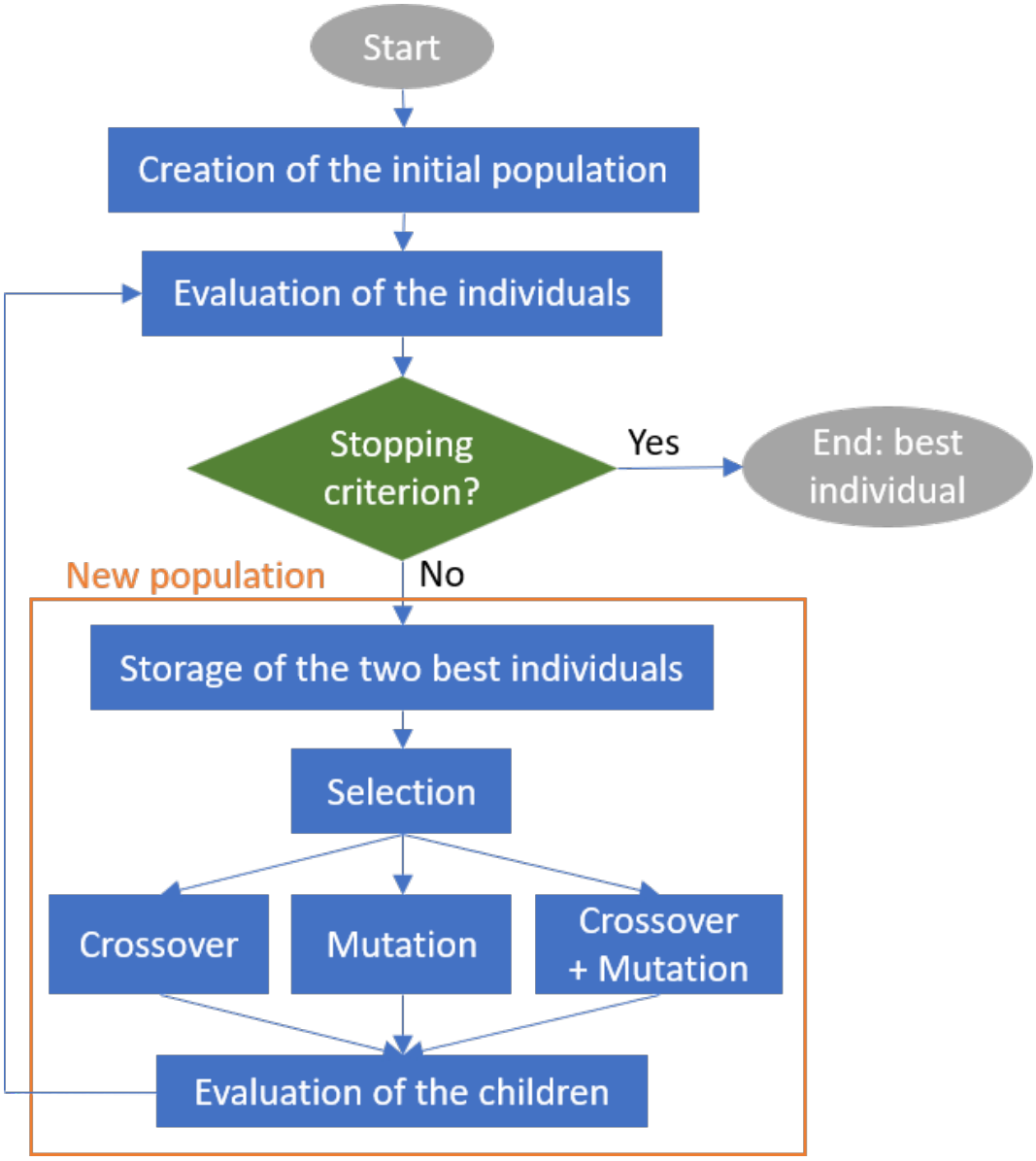


Figure S2. GA procedure.

Table S3. GA parameters.

Parameter	Value	Description
N_{ind}	50	Number of individuals in a population
D_{init}	Depending on the data	Size of the genome for an individual (i.e., number of dimensions or descriptors before dimensionality reduction)
D_{final}	100	Number of 1s in an individual's genome (i.e., number of descriptors to be selected)
N_{gen}	5000 (1000 for benchmark)	Number of generations
α	1	Lasso regularization coefficient
p_c	0.5	Probability for crossover
p_m	0.3	Probability for mutation
p_{c+m}	0.2	Probability for crossover+mutation

S3. Additional results for the enthalpy

S3.1. Data preprocessing

S3.1.1. Elimination of missing descriptor values

Table S4. Effect of the algorithms for the elimination of Desc-MVs on the test *MAE* of the selected ML models for the enthalpy (***preprocessing***: *default*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

Elimination procedure	*MAE* test (kJ/mol)			
	Lasso	SVR lin	ET	MLP
Alg.1: by row	20.4 $\pm$6.4	23.7 $\pm$5.0	39.4 $\pm$8.9	42.9 $\pm$5.0
Alg.2: by column	32.9 $\pm$5.9	32.2 $\pm$4.9	46.3 $\pm$7.8	52.1 $\pm$6.4
Alg.3: alternating row or column	27.6 $\pm$1.7	29.3 $\pm$3.3	42.3 $\pm$5.9	41.1 $\pm$5.5

S3.1.2. Elimination of descriptors with low variance

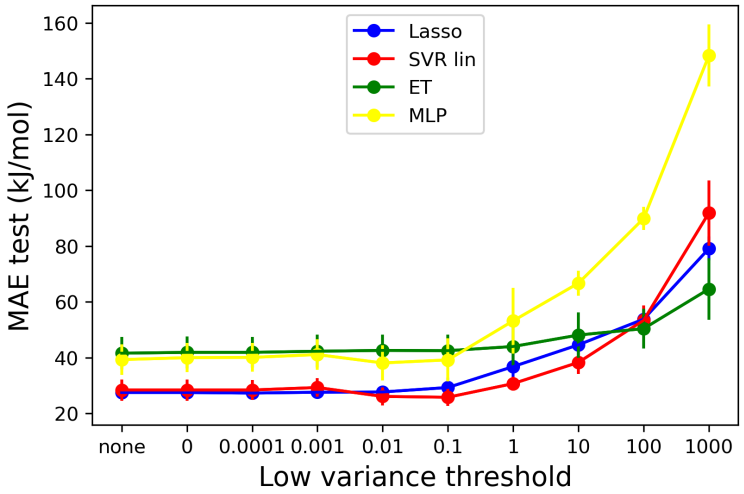


Figure S3. Effect of the value of the variance threshold on the test *MAE* of the selected ML models for the enthalpy (***preprocessing***: *default*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

S3.1.3. Elimination of linearly correlated descriptors

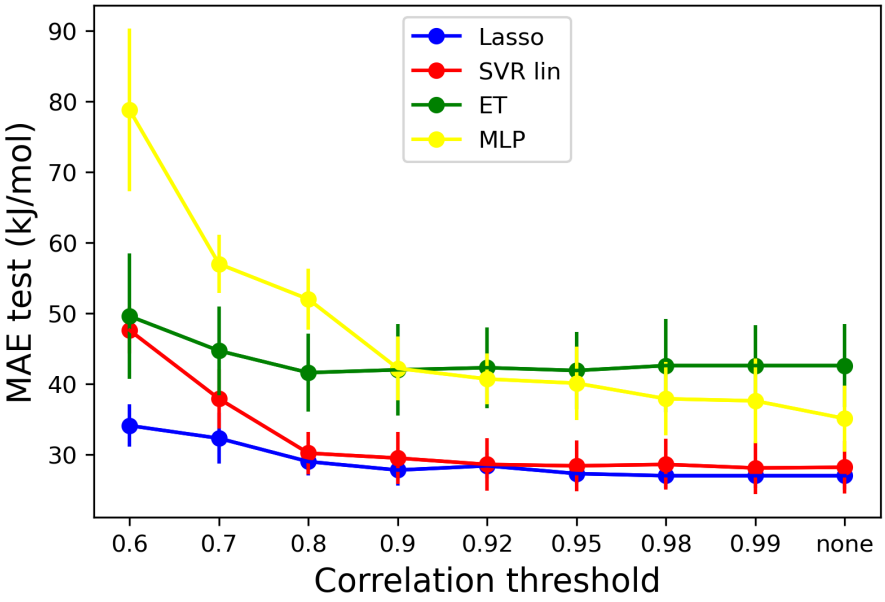
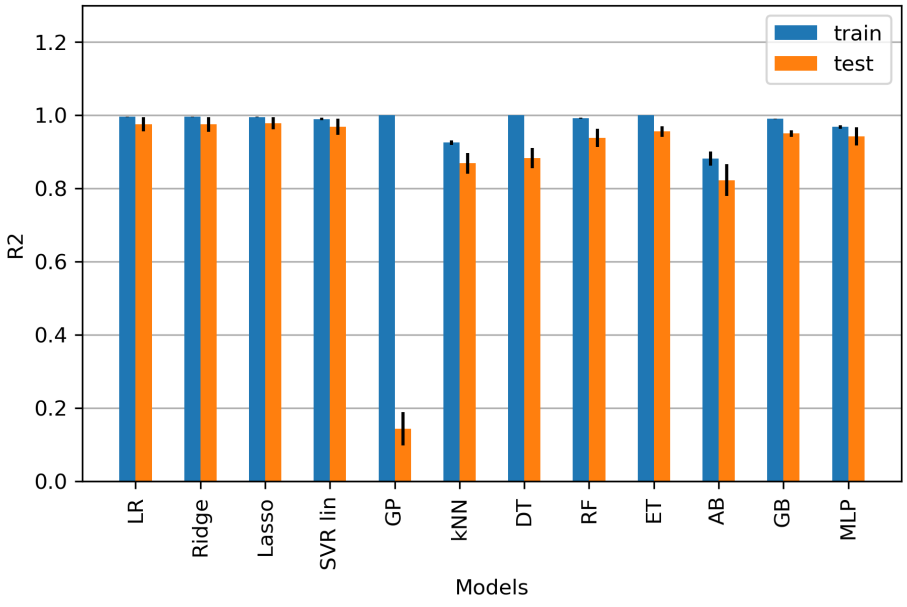


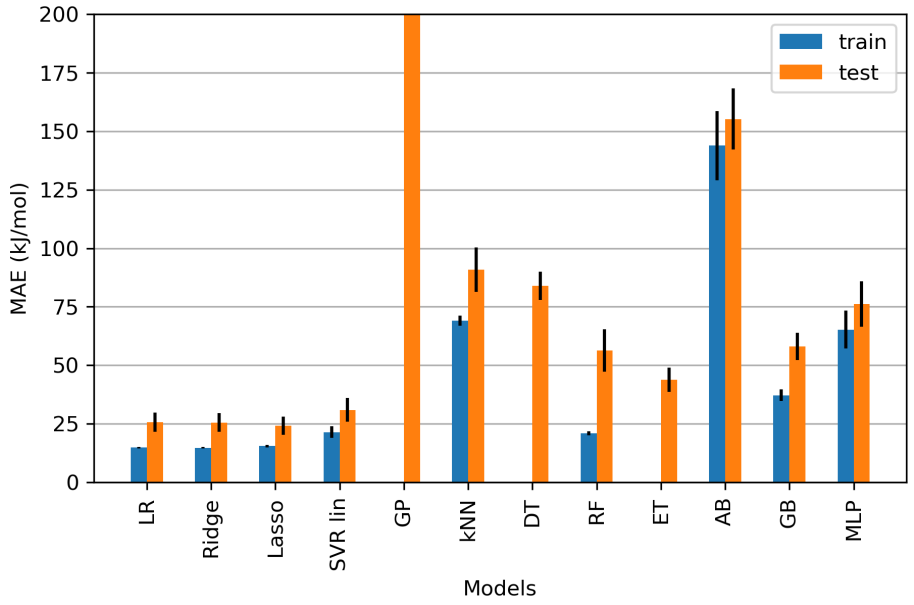
Figure S4. Effect of the value of the correlation coefficient threshold on the test *MAE* of the selected ML models for the enthalpy (***preprocessing***: *default with low variance threshold=0.0001*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

S3.2. Screening with final preprocessing options and wrapper-GA Lasso method for dimensionality reduction

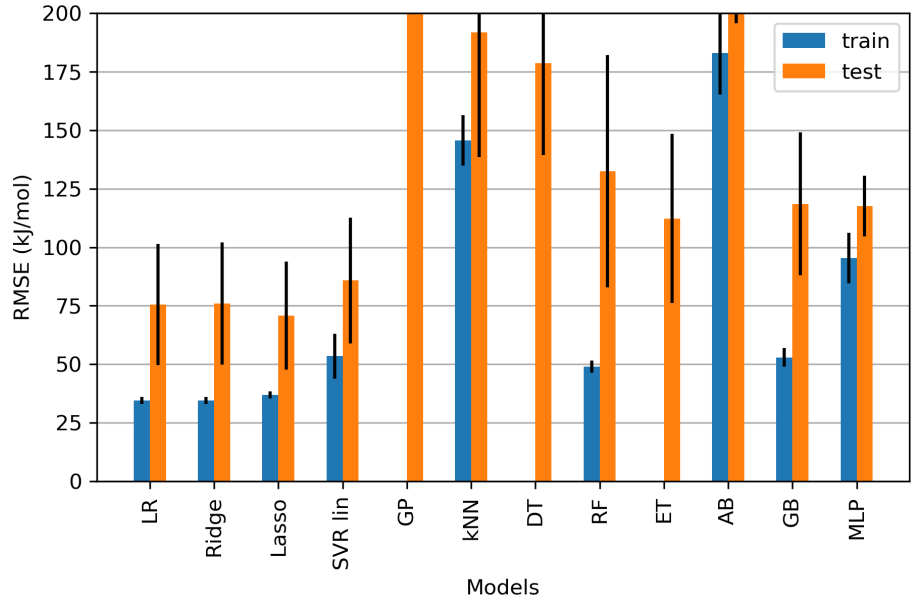


(**a**)

Figure S5. *Cont.*

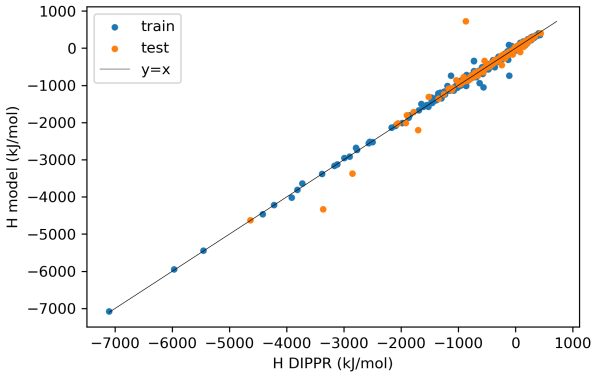


(**b**)

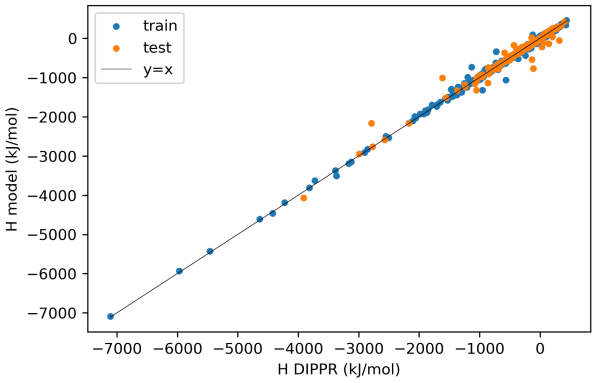


(**c**)

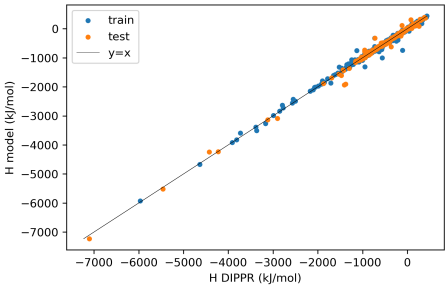
Figure S5. Performance of the different ML models during the final screening for the enthalpy: **(a)** R^2; **(b)** *MAE*; **(c)** *RMSE* (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *none*).



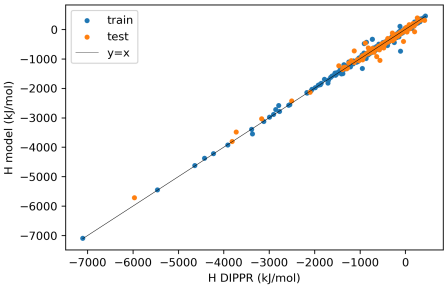
(**a**) Lasso, split 1



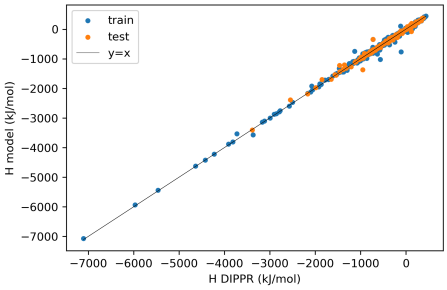
(**b**) Lasso, split 2



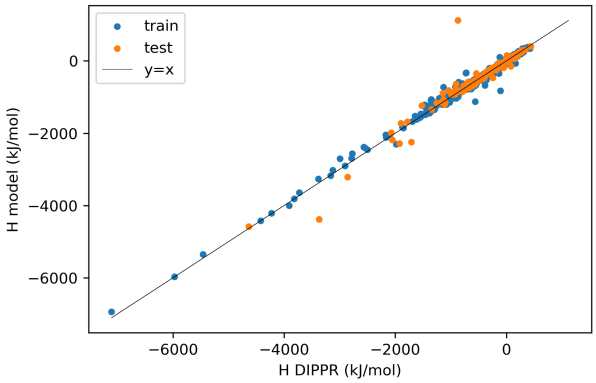
(**c**) Lasso, split 3



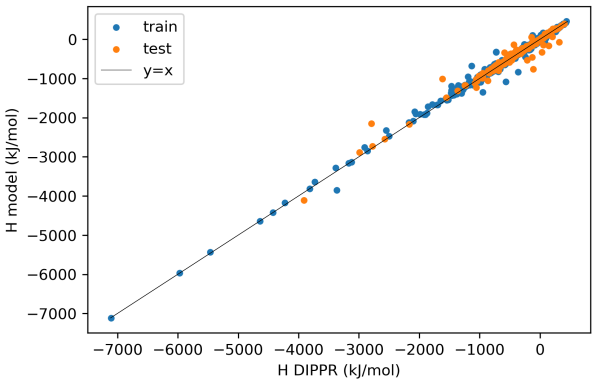
(**d**) Lasso, split 4



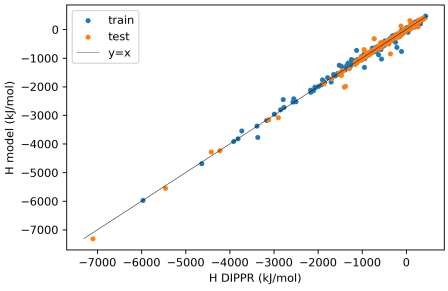
(**e**) Lasso, split 5



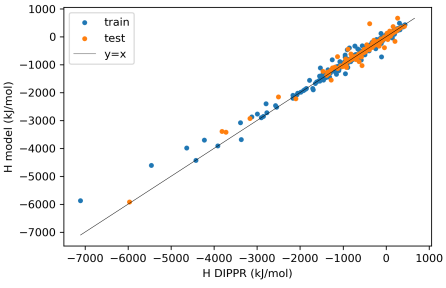
(**f**) SVR lin, split 1



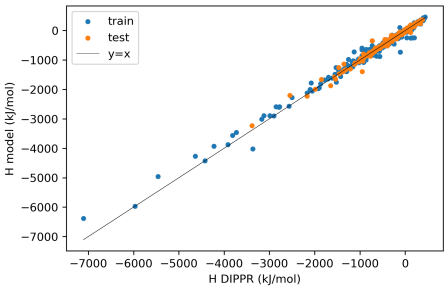
(**g**) SVR lin, split 2



(**h**) SVR lin, split 3

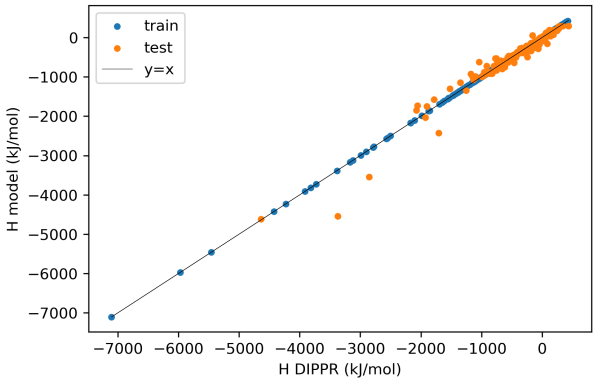


(**i**) SVR lin, split 4

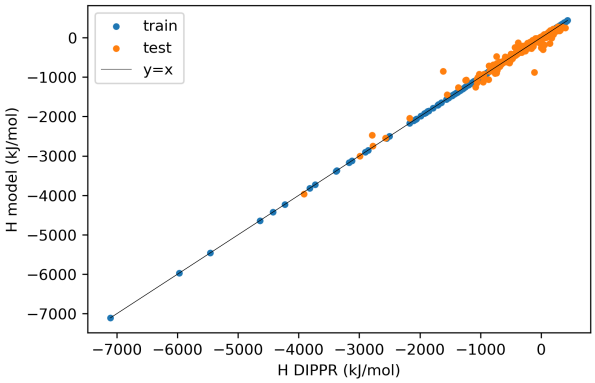


(**j**) SVR lin, split 5

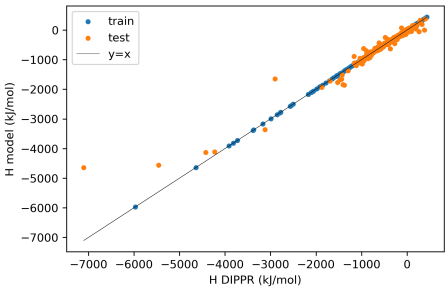
Figure S6. Parity plots of the selected ML models during the final screening, for different splits, for the enthalpy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *none*).



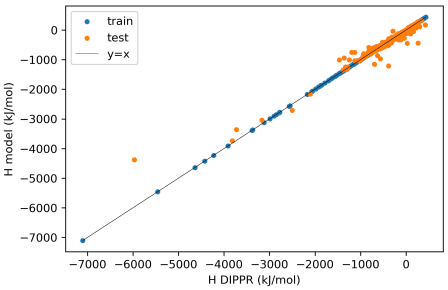
(**a**) ET, split 1



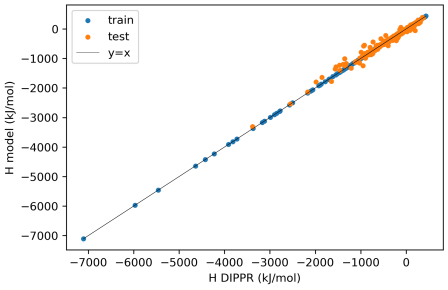
(**b**) ET, split 2



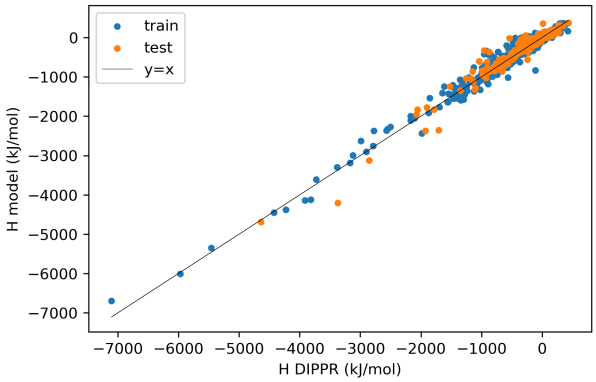
(**c**) ET, split 3



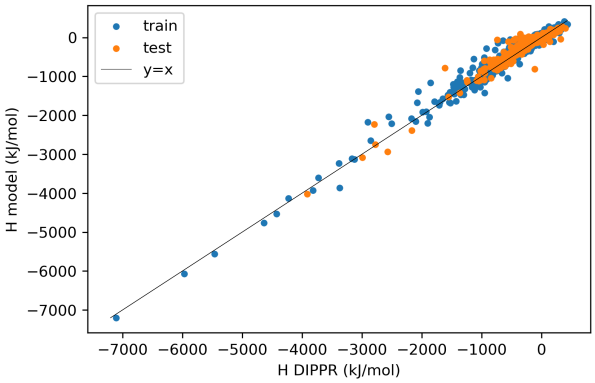
(**d**) ET, split 4



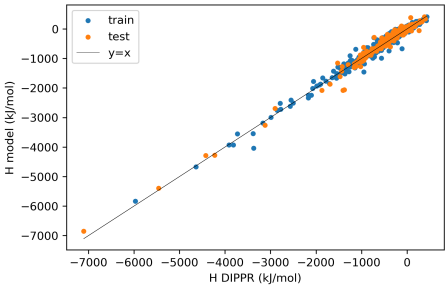
(**e**) ET, split 5



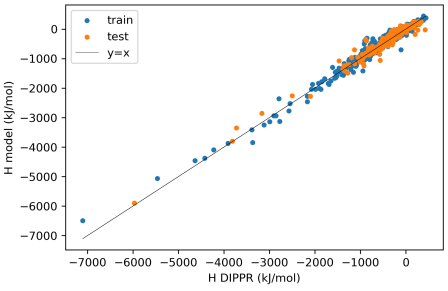
(**f**) MLP, split 1



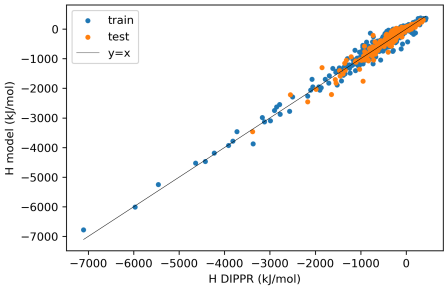
(**g**) MLP, split 2



(**h**) MLP, split 3



(**i**) MLP, split 4



(**j**) MLP, split 5

Figure S7. *Continuation of Figure S6.*

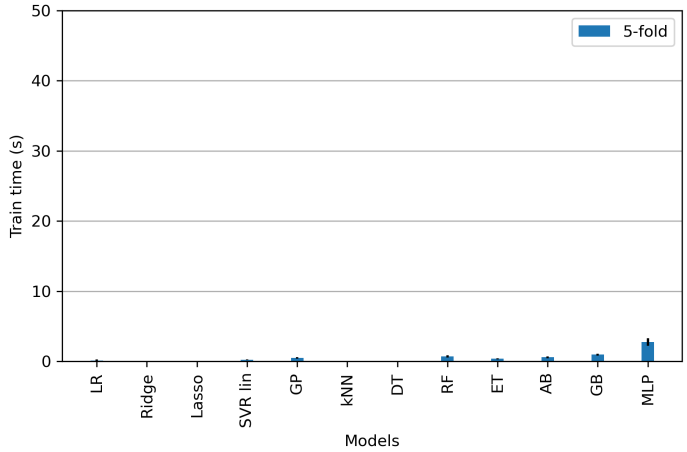
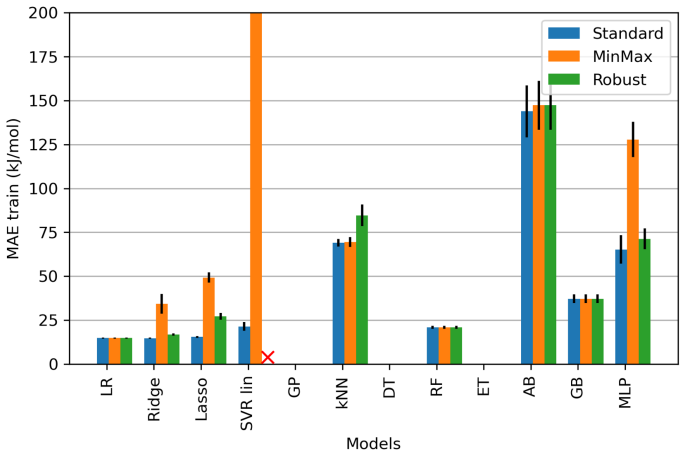
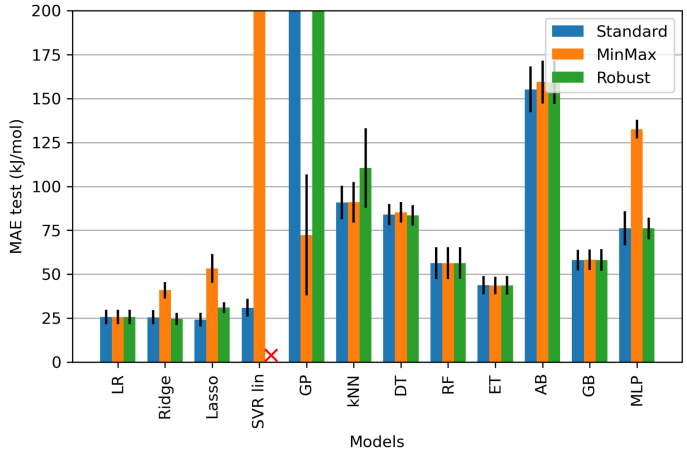


Figure S8. Training time of the different ML models during the final screening for the enthalpy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *none*).



(**a**)

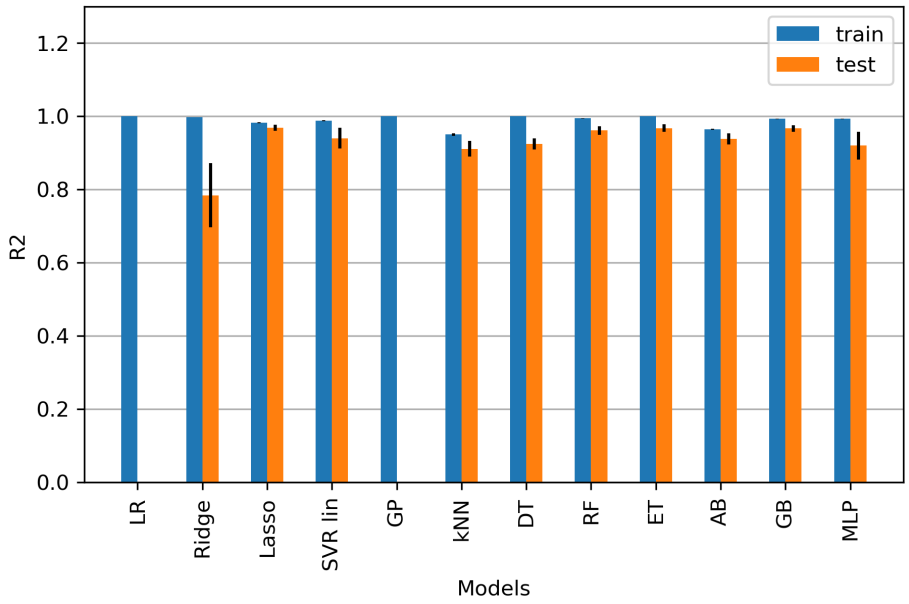


(**b**)

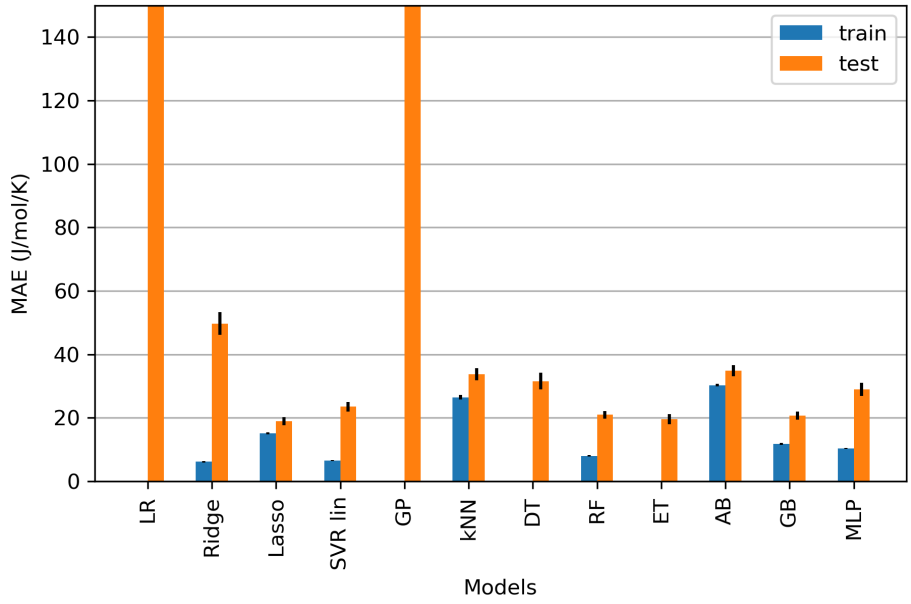
Figure S9. Effect of the data scaling technique on the **(a)** train *MAE* **(b)**, test *MAE*, of the different ML models during the final screening for the enthalpy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *none*). *N.B. Robust scaler did not work with the SVR lin method.*

S4. Results for the entropy

S4.1. Preliminary screening with default preprocessing and without dimensionality reduction

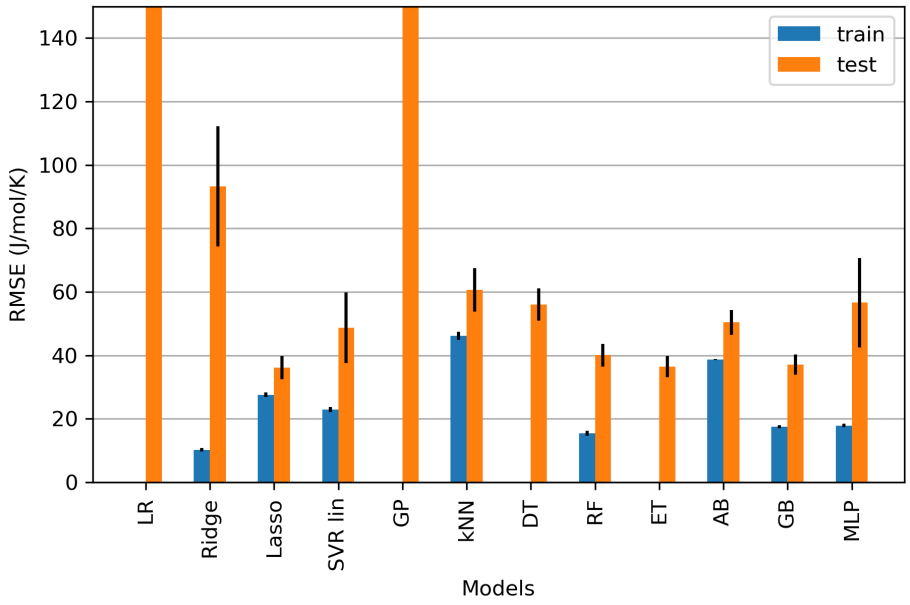


(**a**)



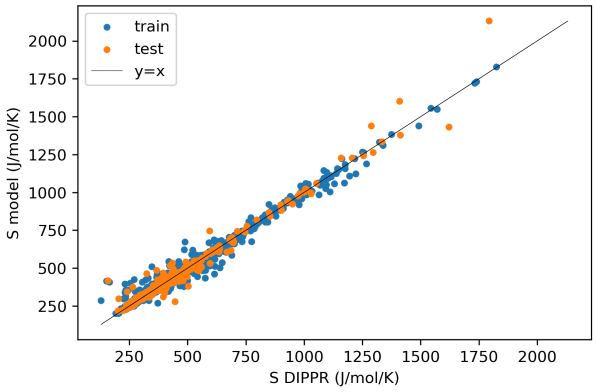
(**b**)

Figure S10. *Cont.*

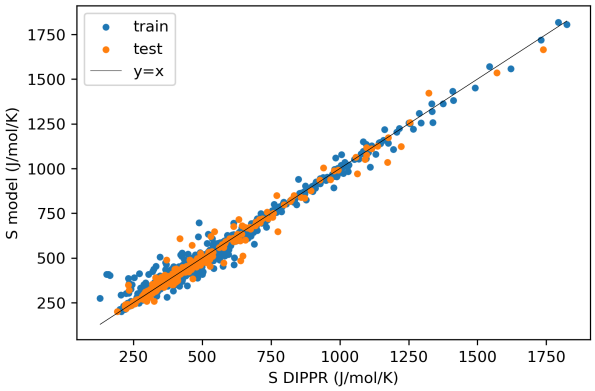


(**c**)

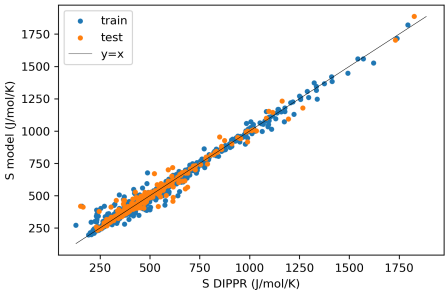
Figure S10. Performance of the different ML models during the preliminary screening for the entropy: **(a)** R^2; **(b)** *MAE*; **(c)** *RMSE* (***preprocessing***: *default*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).



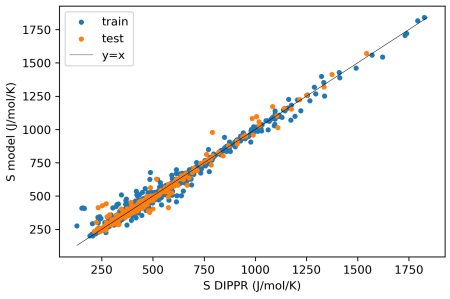
(**a**) Lasso, split 1



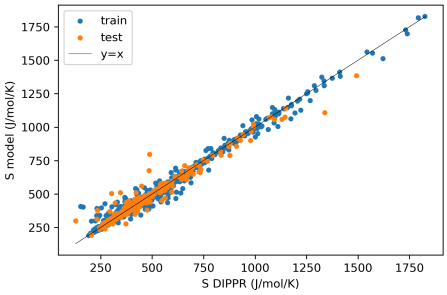
(**b**) Lasso, split 2



(**c**) Lasso, split 3

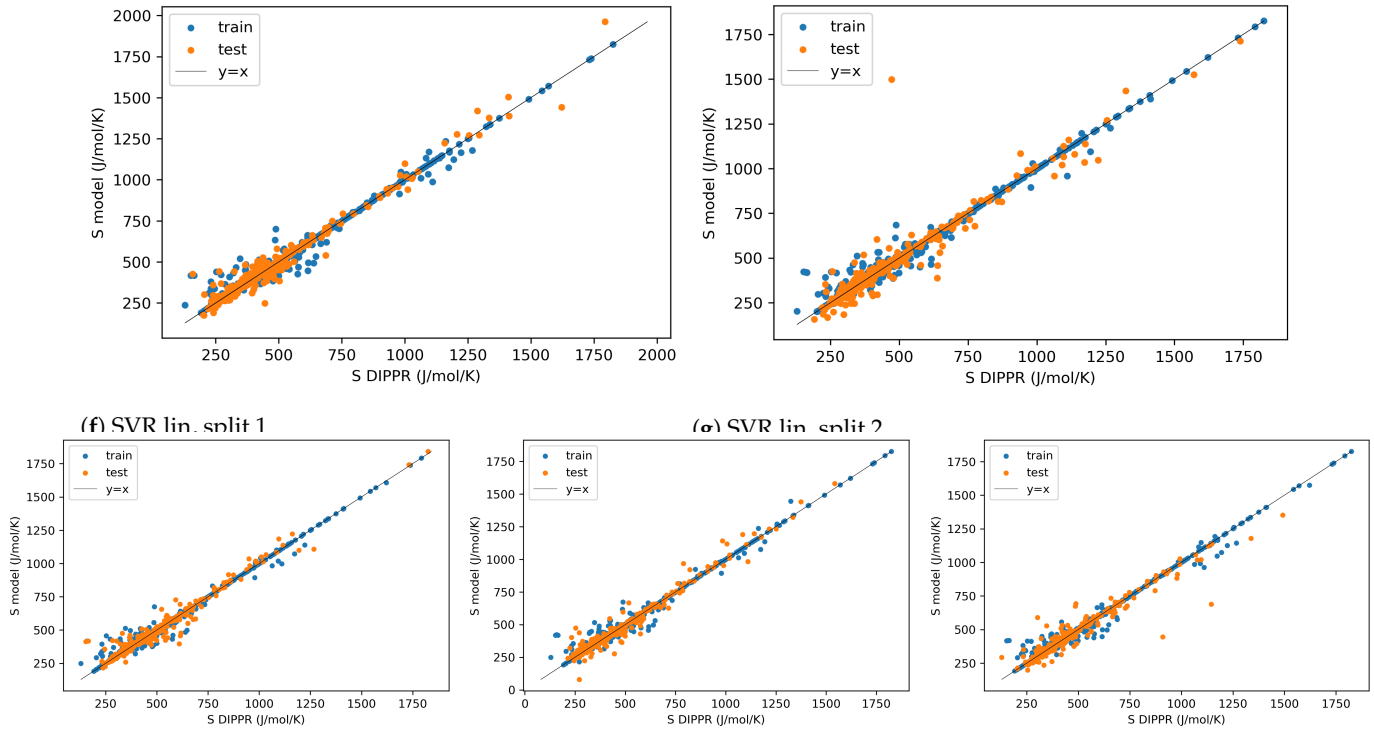


(**d**) Lasso, split 4



(**e**) Lasso, split 5

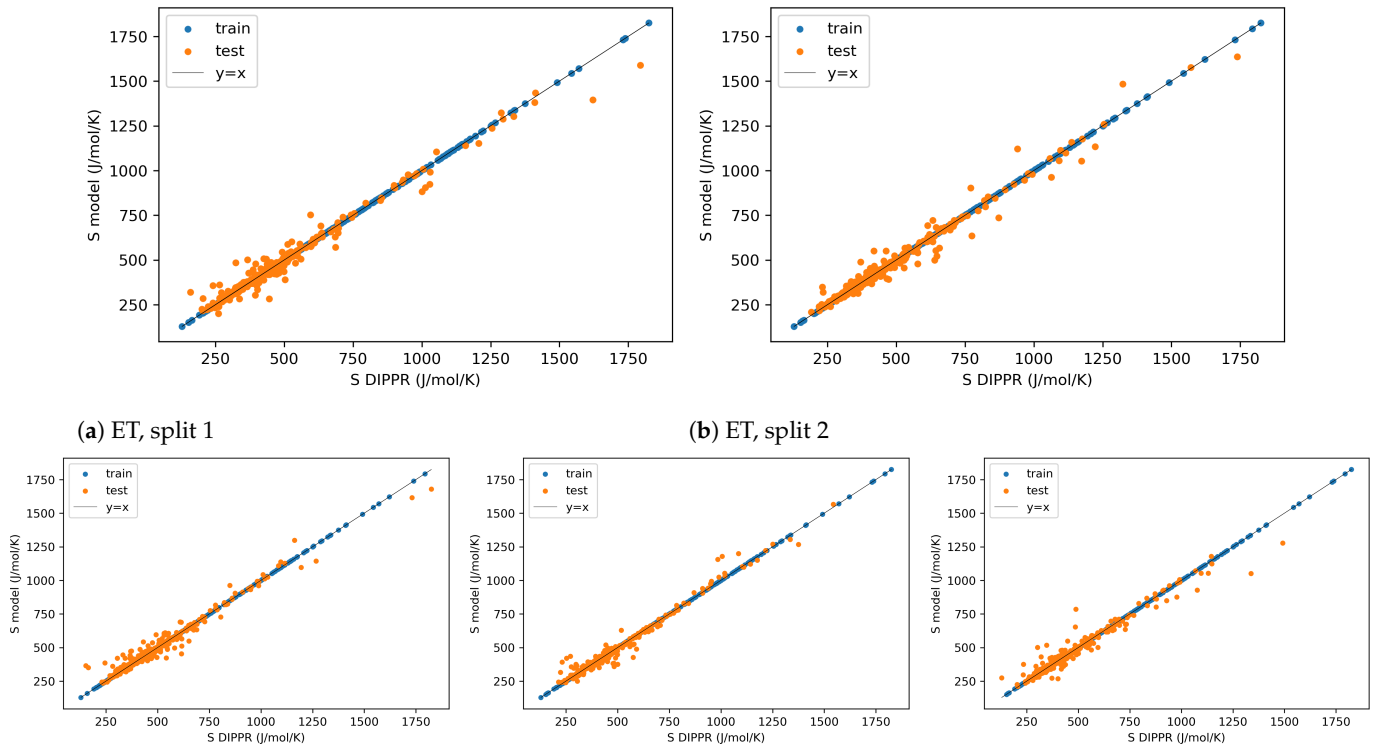
Figure S11. *Cont.*



(**f**) SVR lin, split 1 (**g**) SVR lin, split 2

(**h**) SVR lin, split 3 (**i**) SVR lin, split 4 (**j**) SVR lin, split 5

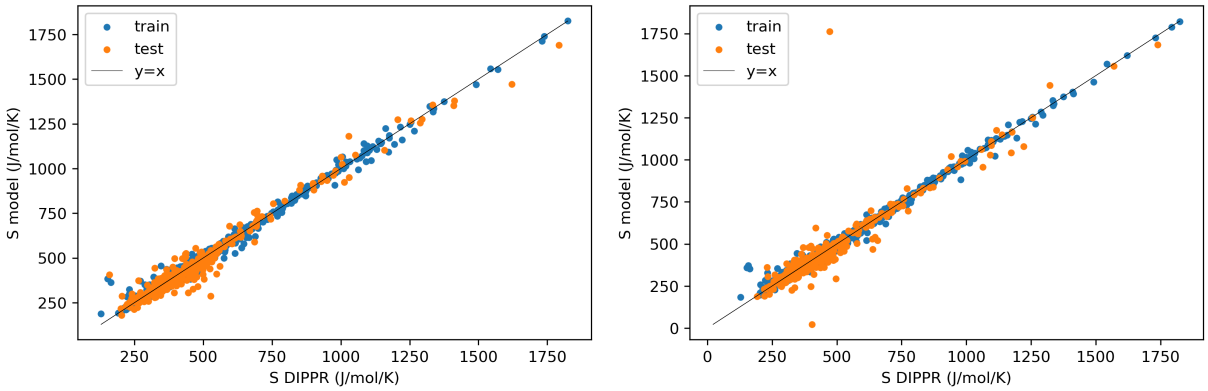
Figure S11. Parity plots of the selected ML models during the preliminary screening, for different splits, for the entropy (***preprocessing***: *default*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).



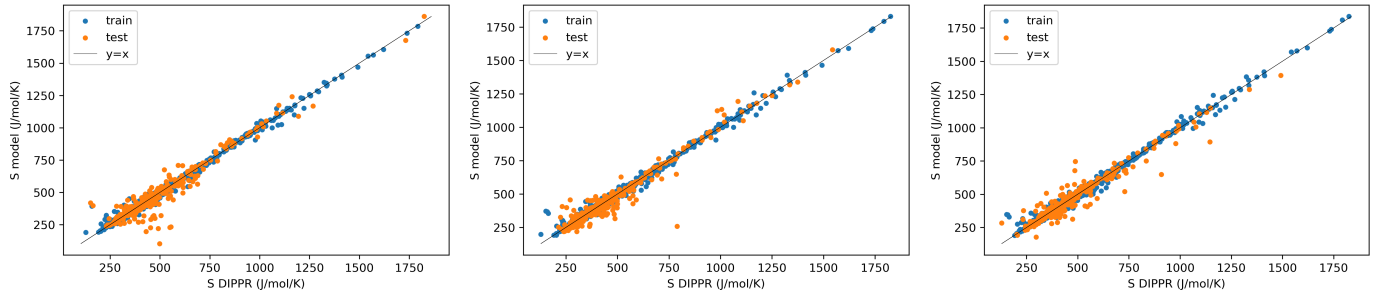
(**a**) ET, split 1 (**b**) ET, split 2

(**c**) ET, split 3 (**d**) ET, split 4 (**e**) ET, split 5

Figure S12. *Cont.*

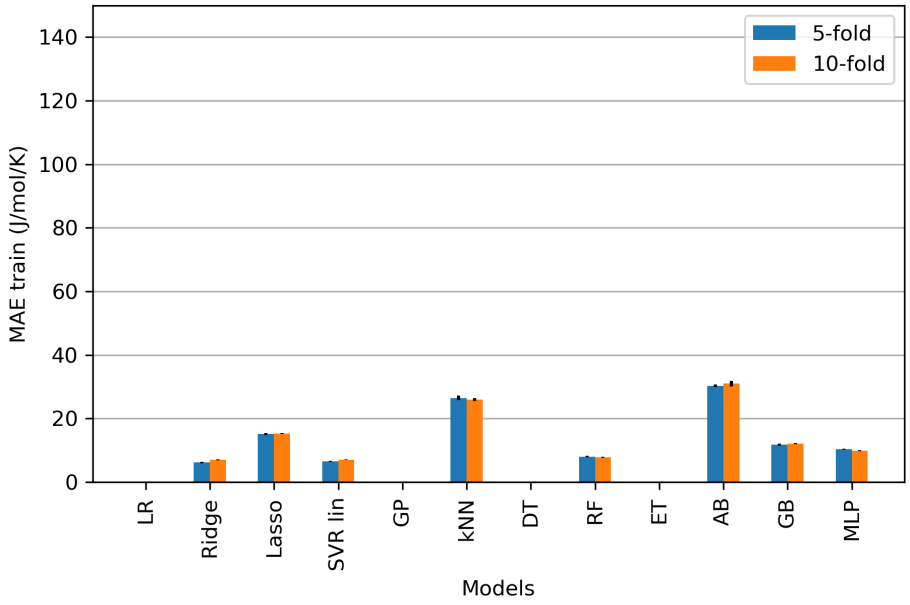


(**f**) MLP, split 1 (**g**) MLP, split 2



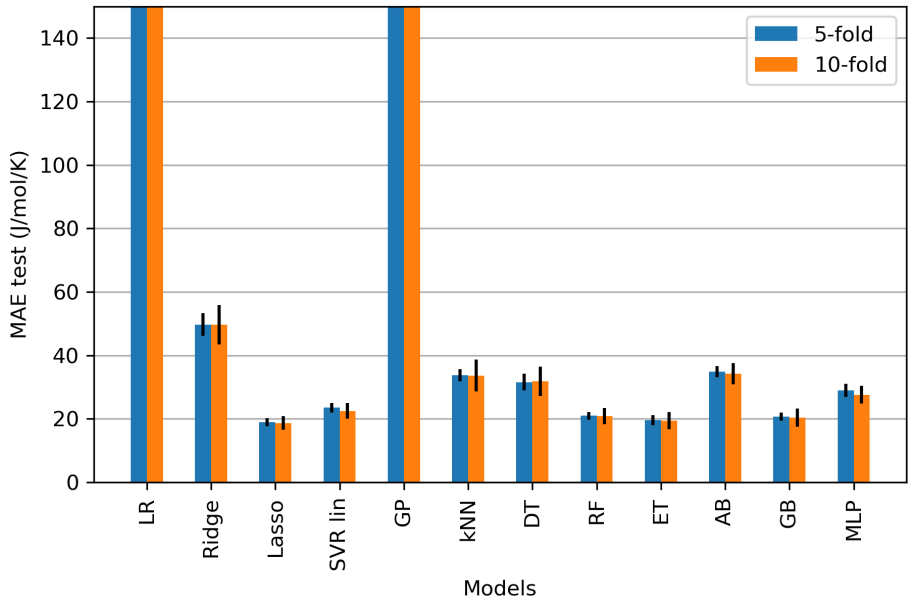
(**h**) MLP, split 3 (**i**) MLP, split 4 (**j**) MLP, split 5

Figure S12. *Continuation of Figure S11.*

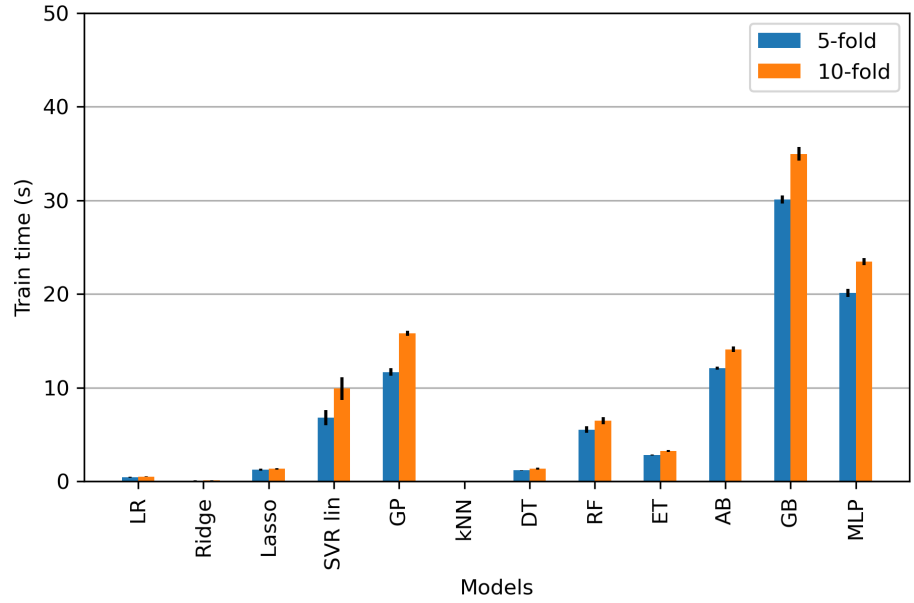


(**a**)

Figure S13. *Cont.*



(**b**)



(**c**)

Figure S13. Effect of the value of k', for the external CV, on the **(a)** train *MAE* **(b)**, test *MAE*, **(c)** training time, of the different ML models during the preliminary screening for the entropy (***preprocessing***: *default*, ***splitting***: *5 and 10-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

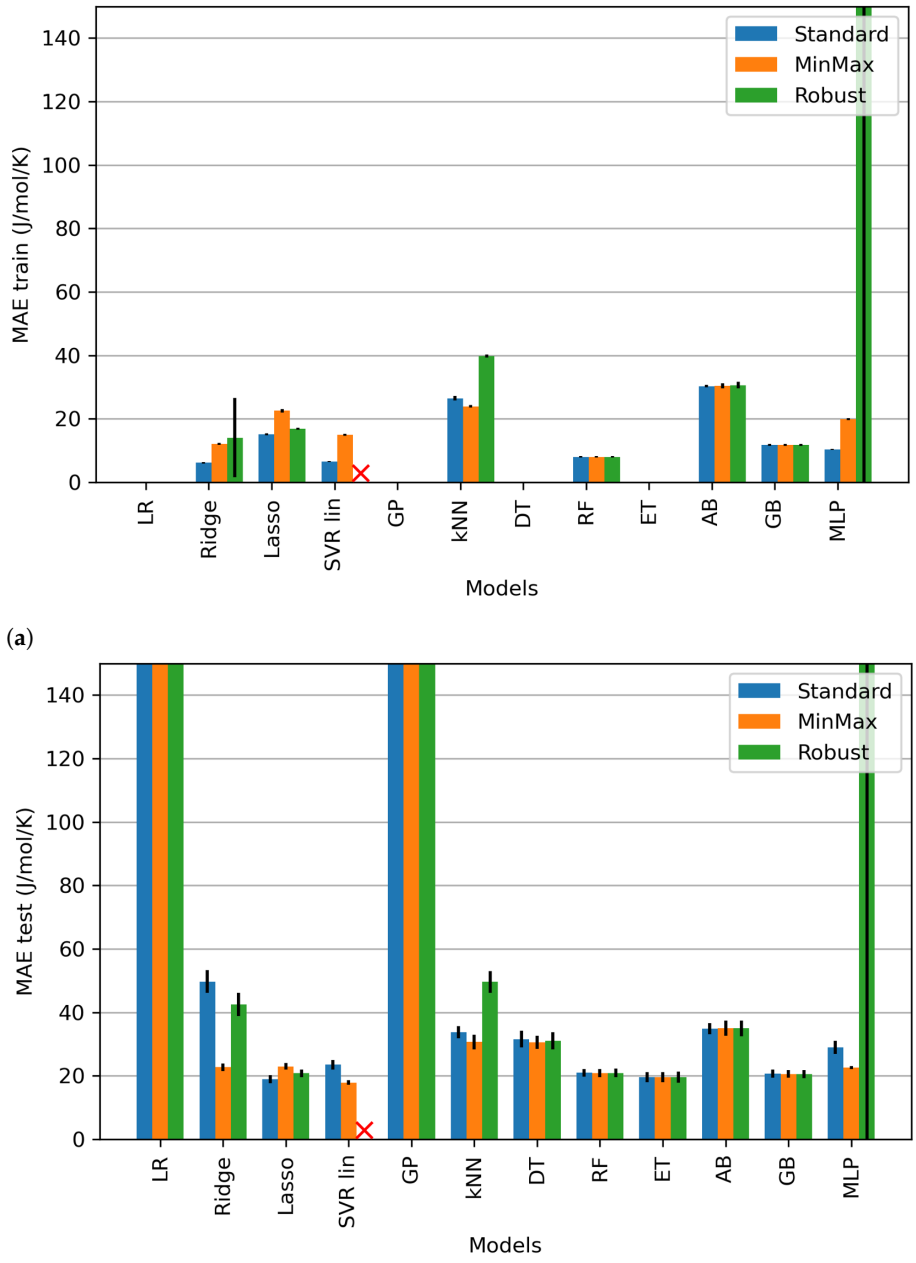


Figure S14. Effect of the data scaling technique on the **(a)** train *MAE* **(b)**, test *MAE*, of the different ML models during the preliminary screening for the entropy (***preprocessing***: *default*, ***splitting***: *5-fold external CV*, ***scaling***: *standard/min-max/robust*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*). *N.B. Robust scaler did not work with SVR lin method.*

S4.2. Data preprocessing

S4.2.1. Elimination of missing descriptor values

Table S5. Effect of the algorithms for the elimination of Desc-MVs on the data set size and Lasso model test *MAE* for the entropy (***preprocessing***: *default*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

Elimination procedure	Data set with Desc-MVs	Data set without Desc-MVs	Data set after preprocessing	*MAE* train (J/mol/K)	*MAE* test (J/mol/K)
Alg.1: by row	1872 x 5666 *mol. desc.*	224 x 5666 *0 duplicates*	224 x 1352	7.6 ±0.3	17.2 ±2.9
Alg.2: by column	1872 x 5666	1872 x 2855 ***75 duplicates***	1872 x 974	16.4 ±0.3	19.7 ±0.9
Alg.3: alternating row or column	1872 x 5666	1747 x 5531 *0 duplicates*	1747 x 1932	15.2 ±0.3	18.9 ±1.3

Table S6. Effect of the algorithms for the elimination of Desc-MVs on the test *MAE* of the selected ML models for the entropy (***preprocessing***: *default*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

Elimination procedure	*MAE* test (J/mol/K)			
	Lasso	**SVR lin**	**ET**	**MLP**
Alg.1: by row	17.2 ±2.9	19.3 ±3.4	17.3 ±2.1	78.3 ±6.8
Alg.2: by column	19.7 ±0.9	21.0 ±2.1	19.7 ±1.1	30.4 ±1.2
Alg.3: alternating row or column	18.9 ±1.3	23.5 ±1.5	19.6 ±1.6	29.0 ±2.1

S4.2.2. Elimination of descriptors with low variance

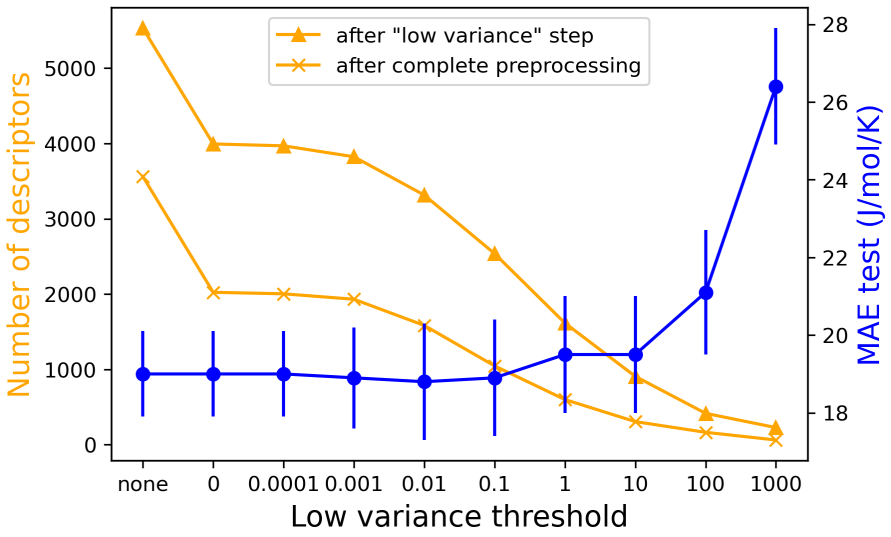


Figure S15. Effect of the value of the variance threshold on the number of retained descriptors and Lasso model test *MAE* for the entropy (***preprocessing***: *default*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

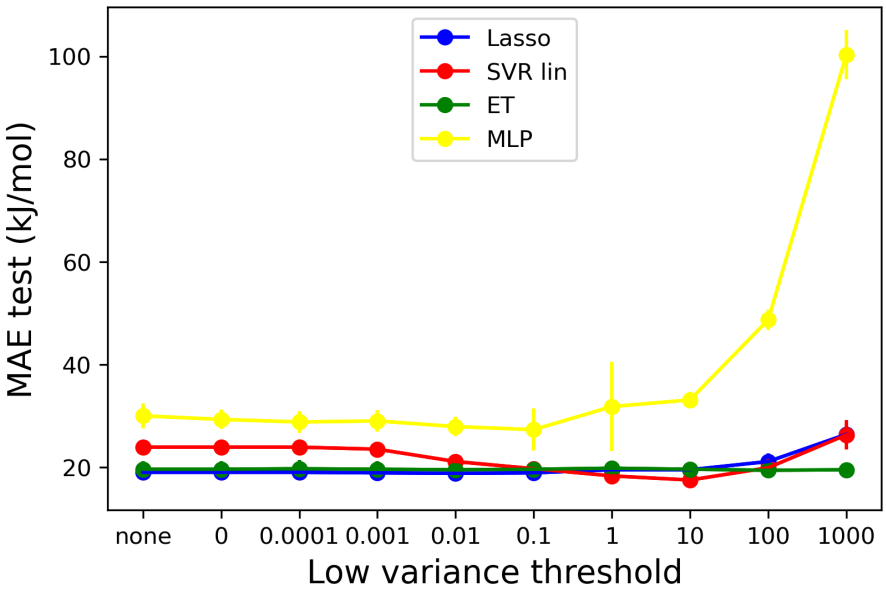


Figure S16. Effect of the value of the variance threshold on the test *MAE* of the selected ML models for the entropy (***preprocessing***: *default*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

S4.2.3. Elimination of linearly correlated descriptors

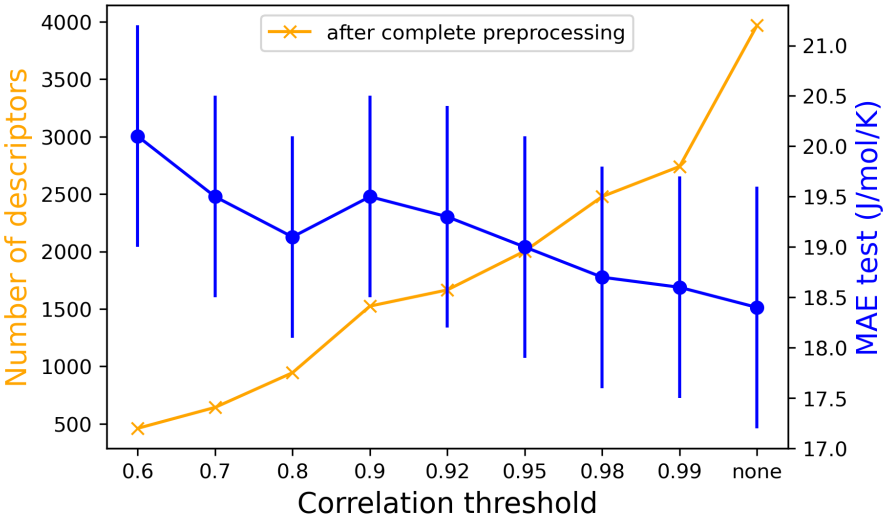


Figure S17. Effect of the value of the correlation coefficient threshold on the number of retained descriptors and Lasso model test *MAE* for the entropy (***preprocessing***: *default with low variance threshold=0.0001*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *none*, ***HP optimization***: *none*).

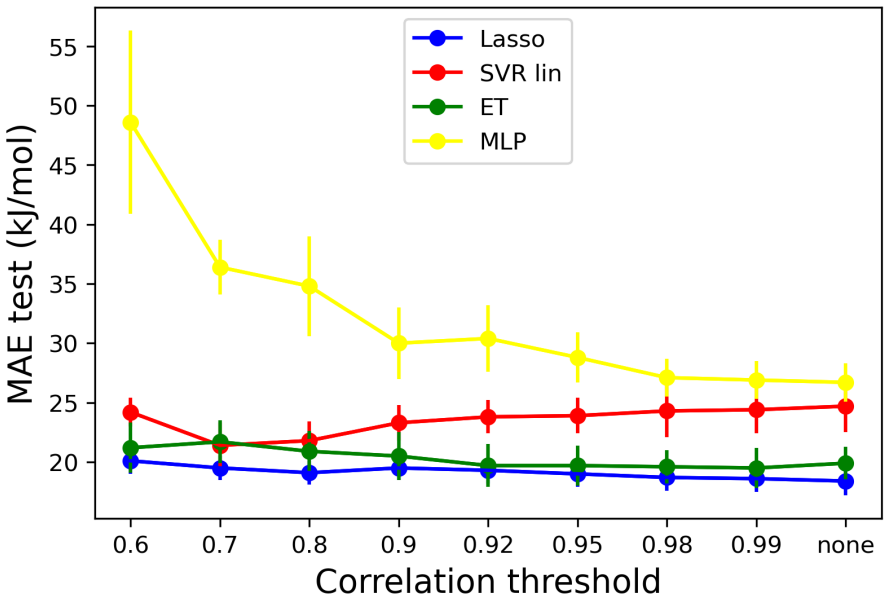


Figure S18. Effect of the value of the correlation coefficient threshold on the test *MAE* of the selected ML models for the entropy (***preprocessing**: default with low variance threshold=0.0001*, ***splitting**: 5-fold external CV*, ***scaling**: standard*, ***dimensionality reduction**: none*, ***HP optimization**: none)*.

S4.3. Dimensionality reduction

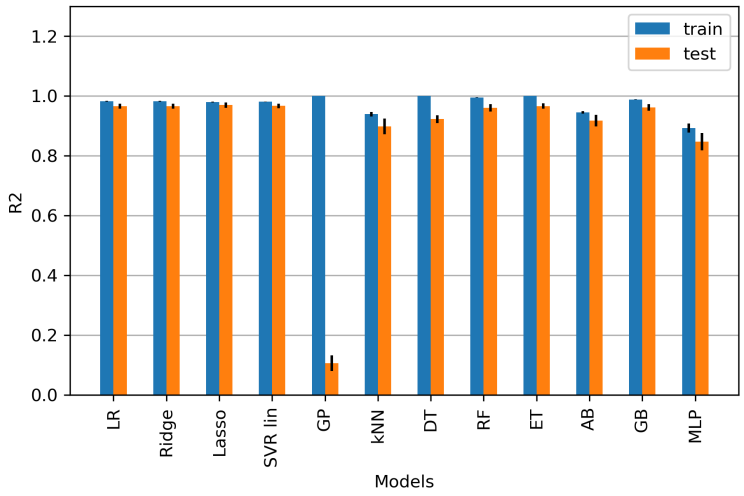
Table S7. Effect of the different dimensionality reduction methods on the test *MAE* of the selected ML models for the entropy (***preprocessing**: final*, ***splitting**: 5-fold external CV*, ***scaling**: standard*, ***dimensionality reduction**: different methods*, ***HP optimization**: none)*.

Dimensionality reduction method	Nb of desc.	Time/ split (s)	*MAE* test (J/mol/K)				Nb of pairwise correlations ≥ 0.9	Nb of desc. with variance ≤ 0.01
			Lasso	SVR lin	ET	MLP		
None (reference case)	2479	0	18.7 ±1.1	24.3 ±2.2	19.6 ±1.4	27.1 ±1.6	4469	487
Filter-Pearson	100	18.1	20.9 ±1.3	18.5 ±1.7	20.3 ±1.3	46.9 ±3.0	909	2
Filter-MI	100	16.3	21.4 ±1.4	19.1 ±1.4	20.0 ±1.6	31.4 ±1.0	514	6
Wrapper-SFS Lasso	100	8294	19.2 ±1.2	18.9 ±1.7	19.6 ±1.9	55.8 ±4.2	15	19
Wrapper-GA Lasso	100	53315	17.4 ±1.2	18.1 ±1.3	19.6 ±1.5	57.5 ±3.7	9	25
Embedded-Lasso	100	1.4	18.8 ±1.1	18.5 ±1.2	19.7 ±1.5	52.6 ±5.6	21	20
Embedded-SVR lin	100	5.6	24.2 ±2.5	23.5 ±2.8	21.3 ±2.0	53.0 ±4.6	8	17
Embedded-ET	100	3.7	21.6 ±1.9	19.7 ±1.4	20.8 ±2.1	31.0 ±1.7	338	8
PCA 95% (254-260 PCs)	2479	3.0	19.7 ±1.0	18.3 ±1.5	23.0 ±1.3	29.1 ±2.3	-	-

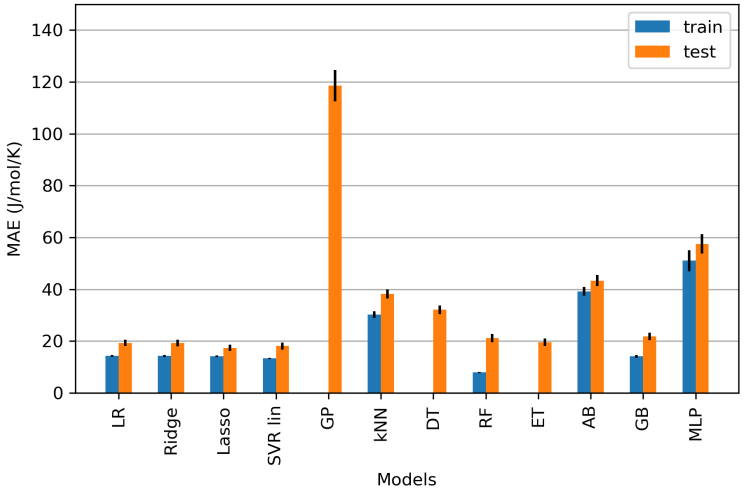
Table S8. Top 5 descriptor categories identified by the different dimensionality reduction methods for the entropy (***preprocessing**: final*, ***splitting**: 5-fold external CV*, ***scaling**: standard*, ***dimensionality reduction**: different methods*, ***HP optimization**: none)*.

Dimensionality reduction method	Top 5 descriptor categories				
None (reference case)	25 12.9%	19 8.7%	8 7.2%	17 6.3%	30 6.2%
Filter-Pearson	7 16.2%	16 15.8%	19 15.4%	3 7.0%	14 7.0%
Filter-MI	7 46.8%	14 10.6%	3 8.4%	8 6.2%	1 4.0%
Wrapper-SFS Lasso	25 14.6%	30 10.4%	21 7.0%	7 6.0%	24 5.4%
Wrapper-GA Lasso	25 16.6%	21 10.4%	30 7.0%	24 5.4%	7 5.0%
Embedded-Lasso	25 17.2%	21 9.6%	16 8.0%	30 7.8%	23 6.0%
Embedded-SVR lin	25 15.8%	30 12.0%	16 8.0%	24 7.4%	21 6.4%
Embedded-ET	7 28.2%	8 14.6%	14 11.8%	9 9.8%	19 8.4%
PCA 95% (254-260 PCs)	25 12.9%	19 8.7%	8 7.2%	17 6.3%	30 6.2%

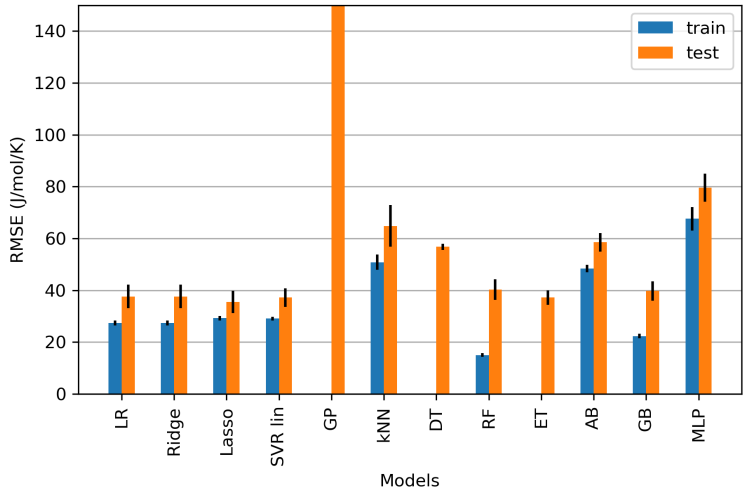
S4.4. Screening with final preprocessing options and wrapper-GA Lasso method for dimensionality reduction



(**a**)

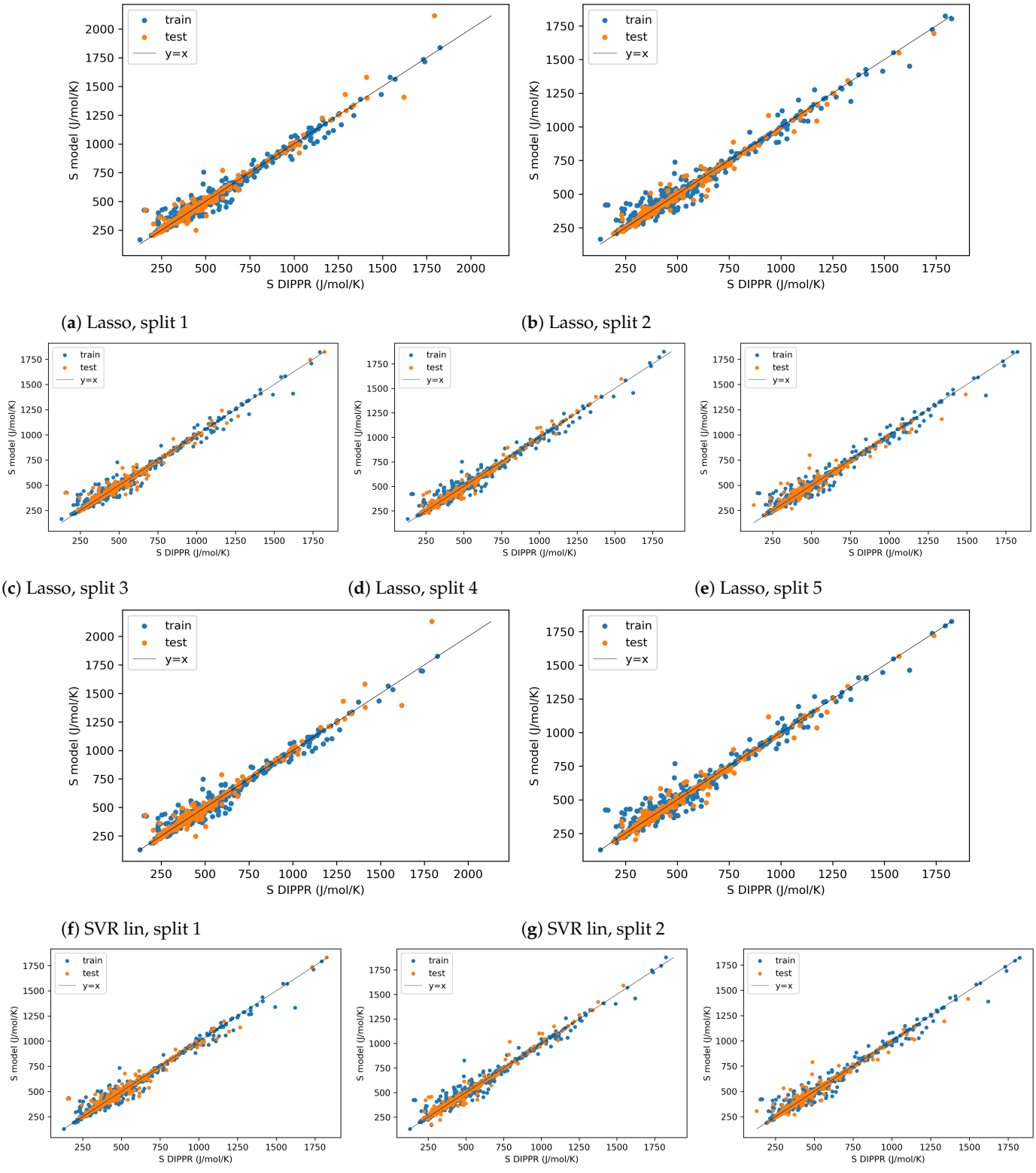


(**b**)



(**c**)

Figure S19. Performance of the different ML models during the final screening for the entropy: **(a)** R^2; **(b)** *MAE*; **(c)** *RMSE* (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *none*).



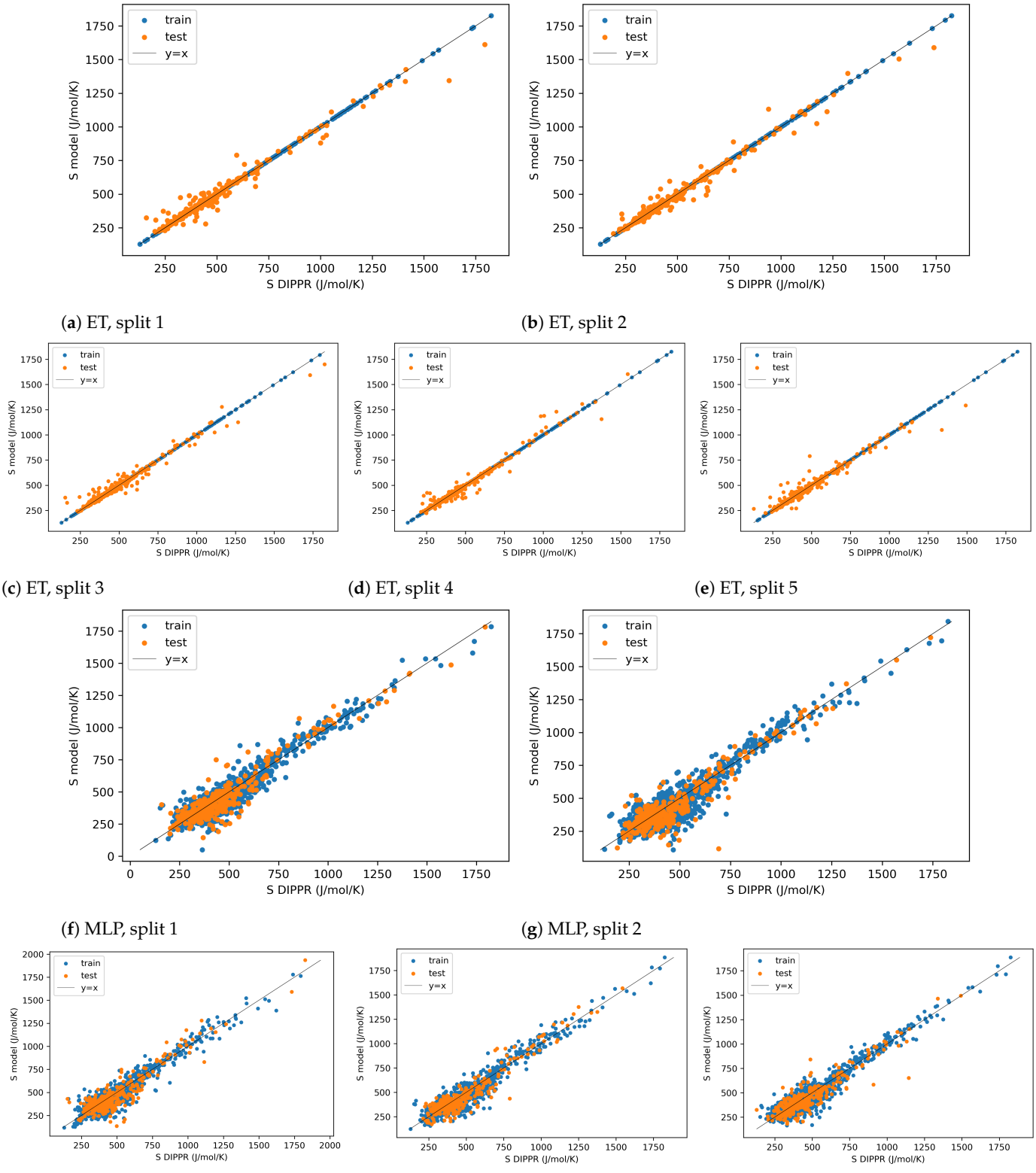
(**a**) Lasso, split 1 (**b**) Lasso, split 2

(**c**) Lasso, split 3 (**d**) Lasso, split 4 (**e**) Lasso, split 5

(**f**) SVR lin, split 1 (**g**) SVR lin, split 2

(**h**) SVR lin, split 3 (**i**) SVR lin, split 4 (**j**) SVR lin, split 5

Figure S20. Parity plots of the selected ML models during the final screening, for different splits, for the entropy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *none*).



(**a**) ET, split 1 (**b**) ET, split 2

(**c**) ET, split 3 (**d**) ET, split 4 (**e**) ET, split 5

(**f**) MLP, split 1 (**g**) MLP, split 2

(**h**) MLP, split 3 (**i**) MLP, split 4 (**j**) MLP, split 5

Figure S21. *Continuation of Figure S20.*

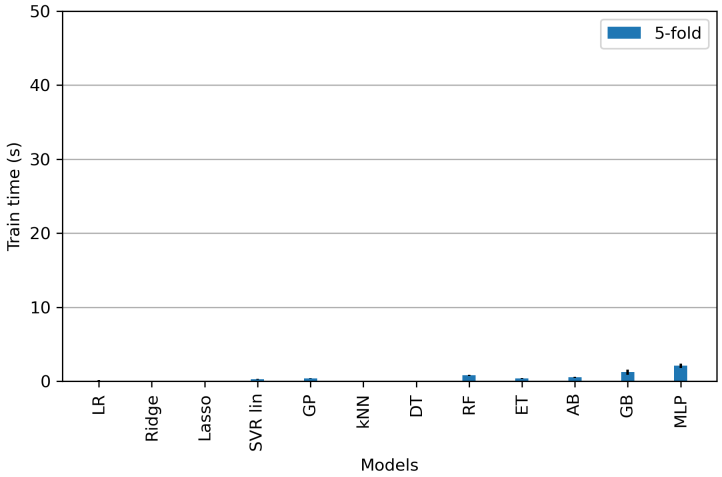
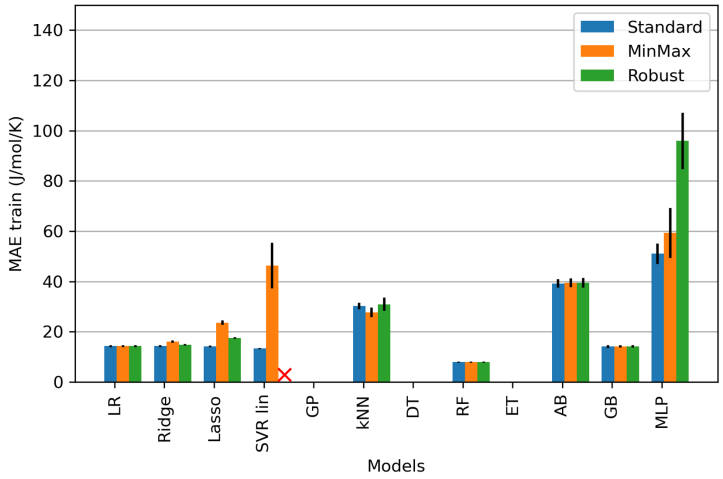
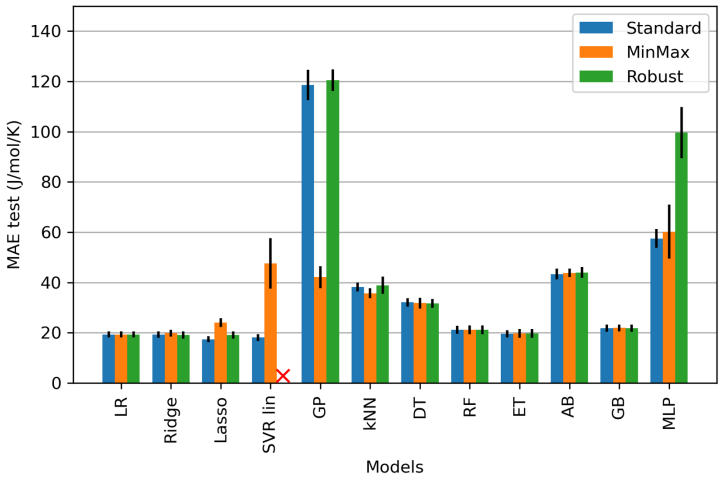


Figure S22. Training time of the different ML models during the final screening for the entropy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *none*).



(**a**)



(**b**)

Figure S23. Effect of the data scaling technique on the **(a)** train *MAE* **(b)**, test *MAE*, of the different ML models during the final screening for the entropy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *none*). *N.B. Robust scaler did not work with the SVR lin method.*

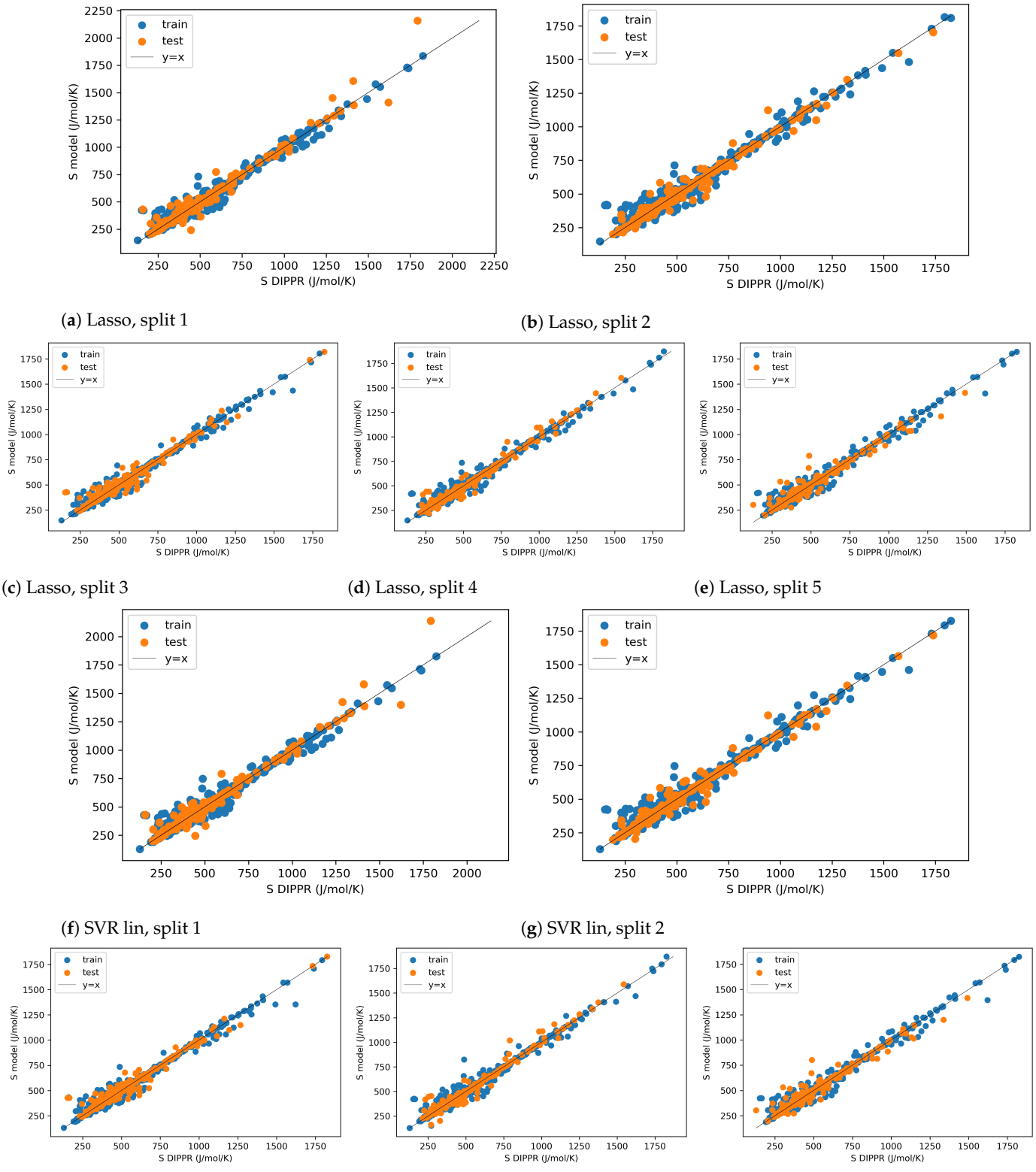
S4.5. HP optimization

Table S9. HP optimization settings and results for the selected ML models for the entropy (***preprocessing**: final*, ***splitting**: 5-fold external CV*, ***scaling**: standard*, ***dimensionality reduction**: wrapper-GA Lasso*, ***HP optimization**: yes*).

ML model	HPs	Screening ranges (blue=default value)	Optimal HP settings per split				
			Split 1	**Split 2**	**Split 3**	**Split 4**	**Split 5**
Lasso	alpha	[0.001, 0.01, 0.1, 0.5, **1**, 1.5, 2]	0.5	0.5	0.5	0.5	0.5
SVR lin	kernel	[**'linear'**]	-	-	-	-	-
	C	[0.1, 0.5, **1**, 1.5, 2]	2	2	2	2	2
	epsilon	[0.01, **0.1**, 1]	0.01	0.01	1	0.1	1
ET	n_estimators	[50, **100**, 200]	100	200	200	200	200
	max_features	['sqrt', 'log2', **None**]	None	None	None	sqrt	None
	min_samples_split	[**2**, 5]	5	5	5	5	2
	min_samples_leaf	[**1**, 5]	1	1	1	1	1
	max_depth	[10, **None**]	10	None	None	None	None
	criterion	['absolute error', **'squared error'**]	absolute	squared	squared	absolute	squared
MLP	activation	[**'relu'**]	-	-	-	-	-
	hidden_layer_sizes	1 hidden layer: [(i)], i = **100**, 200, 400; 2 hidden layers: [(i, i)], i = 10, 15, 20	(15,15)	(10,10)	(15,15)	(10,10)	(15,15)
	solver	[**'adam'**, 'lbfgs']	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
	learning_rate_init	[**0.001**, 0.01, 0.1, 0.5]	0.001	0.001	0.001	0.001	0.001
	max_iter	[**200**, 500]	200	200	200	200	200

Table S10. Performance of the selected ML models with and without HP optimization for the entropy (***preprocessing**: final*, ***splitting**: 5-fold external CV*, ***scaling**: standard*, ***dimensionality reduction**: wrapper-GA Lasso*, ***HP optimization**: none/yes*).

Model	Data set	R^2		*MAE* **(J/mol/K)**		*RMSE* **(J/mol/K)**	
		HP not opt.	**HP opt.**	**HP not opt.**	**HP opt.**	**HP not opt.**	**HP opt.**
Lasso	Train (internal)	0.980 ±0.001	0.982 ±0.001	14.3 ±0.4	13.8 ±0.4	28.8 ±0.8	27.6 ±0.9
	Validation	0.964 ±0.004	0.966 ±0.005	17.9 ±0.6	17.5 ±0.6	35.7 ±1.9	34.5 ±1.8
	Train (external)	0.980 ±0.001	0.982 ±0.001	14.1 ±0.3	13.7 ±0.4	29.2 ±0.9	28.0 ±0.9
	Test	0.969 ±0.009	0.968 ±0.008	17.4 ±1.2	17.9 ±1.2	35.5 ±4.3	36.2 ±4.3
SVR lin	Train (internal)	0.980 ±0.001	0.980 ±0.001	13.2 ±0.3	12.9 ±0.2	28.9 ±0.6	28.7 ±0.7
	Validation	0.970 ±0.002	0.970 ±0.002	16.8 ±0.5	16.4 ±0.4	32.7 ±1.1	32.5 ±1.1
	Train (external)	0.980 ±0.001	0.980 ±0.001	13.3 ±0.2	13.1 ±0.2	29.1 ±0.7	28.9 ±0.7
	Test	0.966 ±0.008	0.966 ±0.008	18.1 ±1.3	17.9 ±1.2	37.2 ±3.6	37.2 ±3.6
ET	Train (internal)	1.000 ±0.000	0.998 ±0.002	0.0 ±0.0	3.9 ±2.7	0.0 ±0.0	8.3 ±5.1
	Validation	0.953 ±0.004	0.953 ±0.004	23.2 ±1.0	22.8 ±0.8	45.7 ±9.0	41.3 ±2.0
	Train (external)	1.000 ±0.000	0.998 ±0.002	0.0 ±0.0	4.0 ±2.9	0.0 ±0.0	8.5 ±5.4
	Test	0.966 ±0.009	0.967 ±0.009	19.6 ±1.5	19.5 ±1.8	37.2 ±2.8	37.0 ±3.7
MLP	Train (internal)	0.848 ±0.026	0.994 ±0.002	60.6 ±5.7	9.8 ±1.0	79.6 ±6.6	16.2 ±2.0
	Validation	0.542 ±0.096	0.929 ±0.008	96.0 ±11.7	31.7 ±1.3	130.3 ±13.0	51.2 ±2.4
	Train (external)	0.892 ±0.015	0.992 ±0.002	51.0 ±4.1	10.8 ±0.8	67.6 ±4.6	17.8 ±1.8
	Test	0.847 ±0.029	0.942 ±0.025	57.5 ±3.7	26.4 ±3.4	79.6 ±5.4	48.3 ±12.0



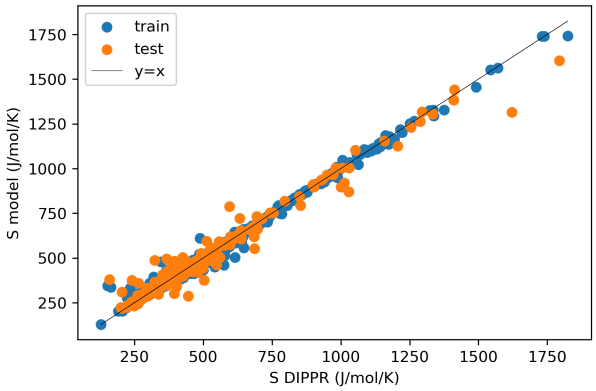
(**a**) Lasso, split 1 (**b**) Lasso, split 2

(**c**) Lasso, split 3 (**d**) Lasso, split 4 (**e**) Lasso, split 5

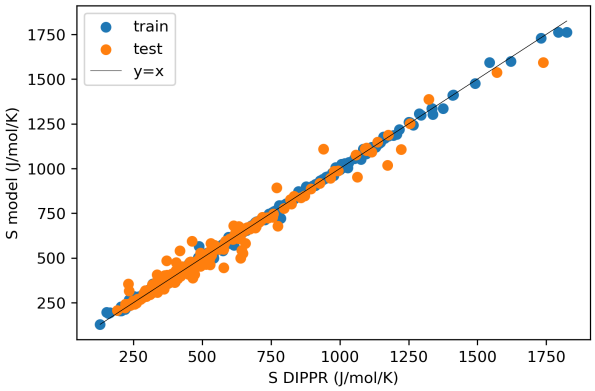
(**f**) SVR lin, split 1 (**g**) SVR lin, split 2

(**h**) SVR lin, split 3 (**i**) SVR lin, split 4 (**j**) SVR lin, split 5

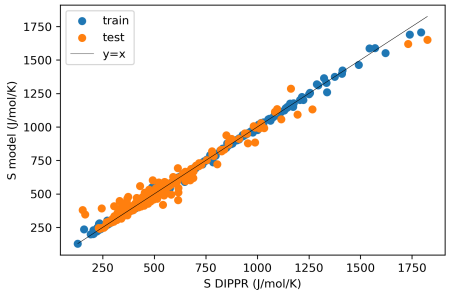
Figure S24. Parity plots of the selected ML models after HP optimization, for different splits, for the entropy (***preprocessing***: *final*, ***splitting***: *5-fold external CV*, ***scaling***: *standard*, ***dimensionality reduction***: *wrapper-GA Lasso*, ***HP optimization***: *yes*).



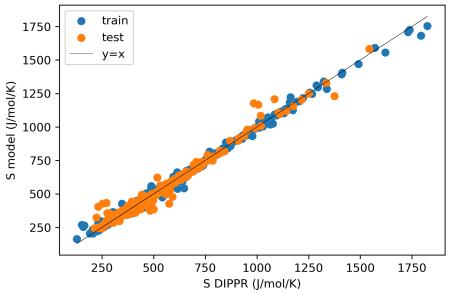
(**a**) ET, split 1



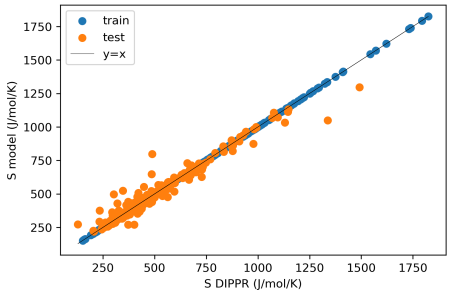
(**b**) ET, split 2



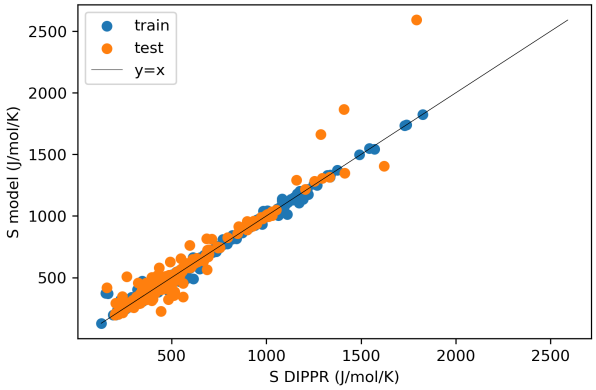
(**c**) ET, split 3



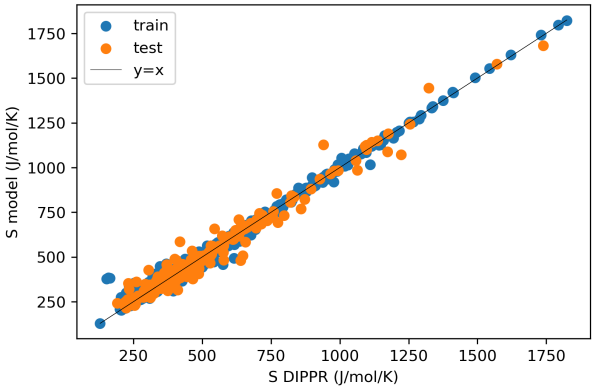
(**d**) ET, split 4



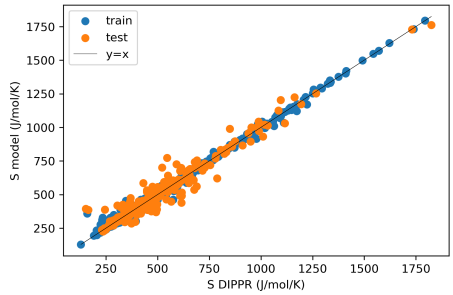
(**e**) ET, split 5



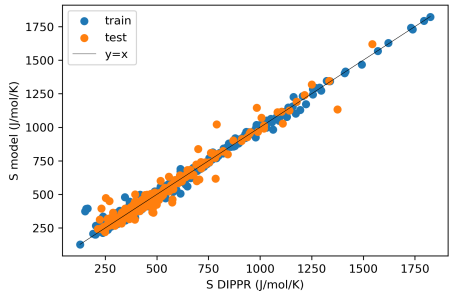
(**f**) MLP, split 1



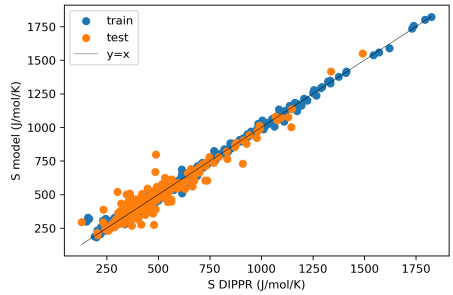
(**g**) MLP, split 2



(**h**) MLP, split 3



(**i**) MLP, split 4



(**j**) MLP, split 5

Figure S25. *Continuation of Figure S24.*

S5. Information on the AlvaDesc descriptor categories identified during dimensionality reduction

Table S11. Description of the most relevant descriptors for the enthalpy (H) and the entropy (S) during dimensionality reduction step, source: AlvaDesc *(P-VSA: Property Van der Walls Surface Area; RDF: Radial Distribution Function; MoRSE: Molecule Representation of Structures based on Electron diffraction; CATS: Chemically Advanced Template Search)*.

Category N°	Category name	Type	Description	Property
7	2D matrix-based descriptors	2D	Descriptors based on different 2D matrices (e.g., adjacency, topological distance, Laplace, Chi matrices)	H and S
10	P_VSA-like descriptors	2D	Based on the sum of atomic contributions to van der Waals surface area, for the atoms having a property in a defined range of values	H
16	RDF descriptors	3D	Probability distribution to find an atom in a given spherical volume	S
17	3D-MoRSE descriptors	3D	3D coordinates are converted to a molecular code with a modified equation used in electron diffraction studies for preparing theoretical scattering curves	H
21	Functional group counts	2D	154 functional group counts	H and S
22	Atom-centred fragments	2D	Structural features initially proposed for the prediction of octanol-water partition coefficient and molar refractivity	H and S
23	Atom-type E-state indices	2D	Information combining the electronic character and the topological environment of the atoms in a molecule	H and S
24	Pharmacophore descriptors	2D	Occurrences of a given topological distance between two given atom types (e.g., hydrogen-bond donor/acceptor). Distribution of pharmacophore features in a molecule.	S
25	2D Atom Pairs	2D	Sums of topological distances between all pairs of atom type. Presence/absence/occurrences of atom pairs (having particular features at a given topological distance).	H and S
30	CATS 3D descriptors	3D	Occurrences of a given Euclidean distance between two given atom types (e.g., hydrogen-bond donor/acceptor)	S

S6. Additional information on the tested tools for descriptor calculation

Two open-source (PaDEL and RDKit) and one closed-source (AlvaDesc) tools were compared for the calculation of descriptors. In this work, AlvaDesc was chosen for several reasons: the high amount of descriptors, the ease of use, the available support/documentation and the robustness of the calculations.

Concerning the amount of available descriptors, AlvaDesc includes 5666 descriptors, while PaDEL and RDKit can calculate 1875 and about 200 descriptors, respectively. In this work, a high amount of descriptors was preferred to increase the chances to capture the relevant structural features for the investigated thermodynamic properties.

Regarding the first open-source tool that was tested, PaDEL, it displayed a lack of robustness, i.e., the descriptors values changed for repeated calculations under identical options. Moreover, the latest release date from 2014 and therefore obtaining support/documentation on the encountered problems is more complex.

As for the second open-source tool, RDKit, the main limitation was the limited amount of descriptors. Except for this, RDKit is a popular toolkit for cheminformatics that has received a lot of contributions from RDKit open-source community and therefore benefits from continuous developments (releases every 6 months). RDKit is well-documented and can be easily used, even if some programming knowledge is required (contrary to AlvaDesc and PaDEL). On top of descriptor calculation, RDKit can also be used for 2D and 3D molecular operations (e.g., substructure searching, chemical transformations, molecular similarity), which makes it an interesting multi-functional tool when working with molecules.

More descriptor calculation tools can be found in the literature and the work of [60] provides a comparison between more descriptor calculation tools (e.g., Mordred, BlueDesc, ChemoPy, Rcpi and Cinfony).

Chapter 3

Descriptor-based QSPR/QSAR models: the applicability domain problem in high dimension

Contents

3.1 Introduction and summary of the publication

The final objective of developing QSPR models is to be able to apply them for the prediction of the properties of interest of new molecules, for which a reference base or measurement does not necessarily exist. In the pursuit of establishing this predictive capacity, many questions are generated regarding the indicators that will allow assessing the reliability of a new prediction.

To determine whether the model developed in Chapter 2 can be applied to a given new molecule, the model's applicability domain (AD) is investigated in the following chapter. The AD contains the set of chemical information (including structures and properties) for which a model is capable of making predictions with a certain reliability. In fact, knowing if a ML model has extrapolated or not improves the interpretability of the model, which is increasingly requested, and therefore brings more transparency and acceptance to the model [10].

In general, most existing methods for defining the AD are more suitable for low-dimensional applications (i.e., less than 10 dimensions, whereas the developed models contain 100-1000 dimensions/descriptors). Besides, these methods are generally used during model construction or deployment stages to check the location of validation/test/new data with respect to the AD defined by training data. Note that the term "new" here refers to the data that is not used for model construction (i.e. training and validation of the model). In this chapter, different methods, more suitable for high-dimensional data sets, are investigated at different stages of the QSPR model, as presented in Figure 3.1.

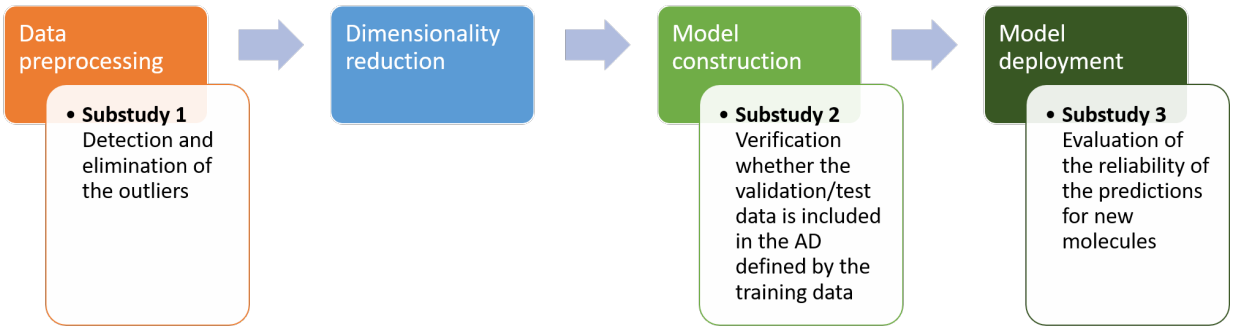


Figure 3.1: Overview of the methodology adopted in Chapter 3.

In fact, within the different stages of a ML model development procedure, the AD has a specific function. As mentioned earlier, during model construction and deployment, the visualization of the AD enables to verify if any data is inside or outside the AD defined by the training data. The differences between these two stages are the type of data being verified (i.e., validation/test data during model deployment; new data during model deployment) but above all the usage/conclusions that will be made from the analysis. During model construction, knowing the location of validation and test data will allow to evaluate the generalization capability of the models inside or outside the AD. During model deployment, the location of a new molecule inside or outside the AD will give information on the reliability of the predicted property. This work also investigates the use of the AD during data preprocessing stage. In this stage, the visualization of the AD enables to detect eventual outliers and study the impact of keeping or removing them. Indeed, as the AD is directly related to the training data, its shape (and therefore chemical information content) is already determined in the early stages of ML model

development procedure, i.e., during data preparation (collection and preprocessing).

This publication investigates the AD of the previously developed models for the enthalpy of formation and the entropy (cf. Chapter 2) and is structured as follows. First, a discussion on the classical methods for AD definition is provided, while highlighting their lack of compatibility with high-dimensional problems. At the same time, examples of methods for AD definition in high-dimensional problems are also presented. Then, the AD domain is investigated at different stages of ML model development procedure: during data preprocessing, during model cosntruction and during model deployment. In particular, the employed methods and results are described and discussed.

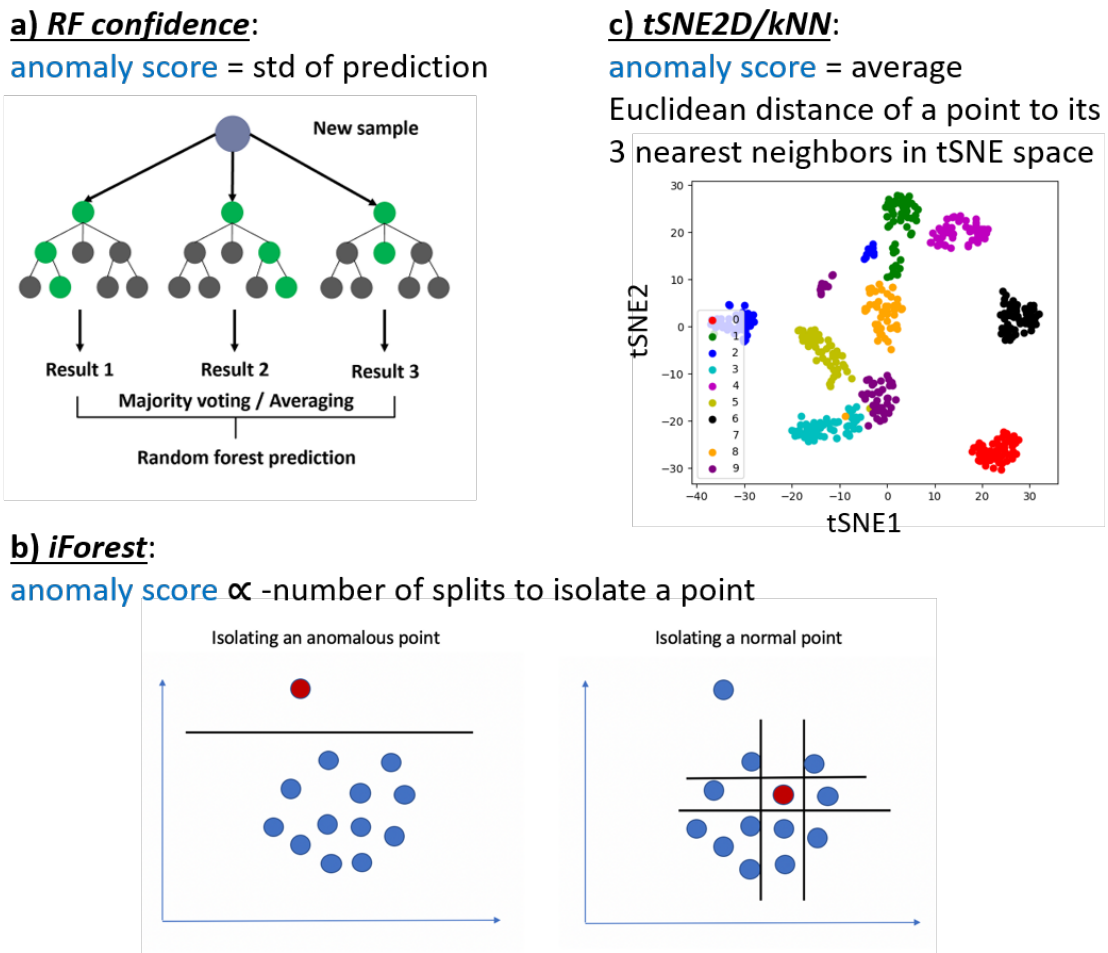


Figure 3.2: Investigated methods for AD definition in high dimension: **(a)** *RF confidence*, **(b)** *iForest*, **(c)** *tSNE2D/kNN* [1, 2, 4].

Three AD definition methods, commonly used for outlier detection in high-dimensional problems, are compared: isolation forest (*iForest*), random forest prediction confidence (*RF confidence*) and k-nearest neighbors in the 2D projection of descriptor space obtained via t-distributed stochastic neighbor embedding (*tSNE2D/kNN*). These methods compute an anomaly score that can be used instead of the distance metrics of classical low-dimension AD definition methods (cf. Figure 3.2), these latter being generally unsuitable for high-dimensional problems. Typically, in low- (high-) dimensional problems, a molecule is considered to lie within the AD if its distance

to the training domain (anomaly score) is below a given threshold.

In this work, due to the high number of descriptors, the AD is graphically represented following the template shown in Figure 3.3, and referred to as "AD plot". The x-axis corresponds to the anomaly score calculated via the different techniques more compatible with high-dimensional data (cf. previous paragraph), based on the descriptor matrix X. This x-axis enables to identify X-outliers, which are outliers in the descriptor space X. On the y-axis, the standardized residuals, which represents the model prediction errors, are considered. The combination of x- and y- axis permits to visualize the chemical structure space within which a model can make predictions with a given reliability. In other words, it shows how well predicted are the properties of the molecules structures based on their anomaly score. In case of poor prediction, the concerned molecules can be considered as Model-outliers. Finally, a color bar can be used to visualize the property values of DIPPR data, referring to the "response space" in the AD definition and highlighting eventual y-outliers (the points with dark circles in Figure 3.3 are y-outliers according to the interquartile range method applied to the response space y). There are no exact rules to define the frontiers of the AD in this plot (i.e., thresholds for anomaly score, standardized residuals and response value), even if some rules-of-thumb can be used (i.e., standardized residuals between -3 and 3).

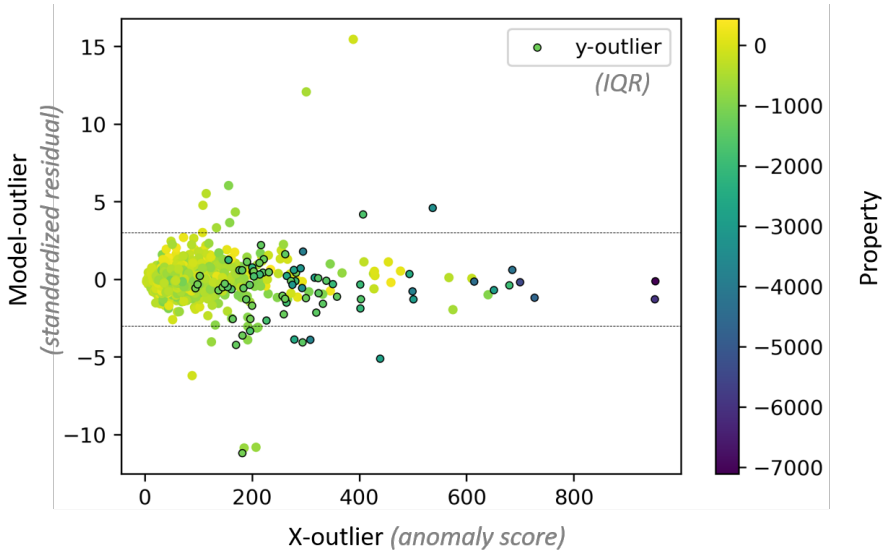


Figure 3.3: AD plot employed in this study. The anomaly score on the x-axis is computed via different methods: *iForest*, *RF confidence* or *tSNE2D/kNN*.

The main results of this chapter are summarized in the following. During data preprocessing, the three AD definition methods are used to identify outlier molecules and the effect of their removal is investigated. A more significant improvement of model performance is observed when outliers identified with *RF confidence* are removed (e.g., for a removal of 30% of outliers, the MAE (Mean Absolute Error) of the test data set is divided by 2.5, 1.6 and 1.1 for *RF confidence*, *iForest* and *tSNE2D/kNN* respectively). While these three methods identify X-outliers, the effect of other types of outliers, namely Model-outliers and y-outliers, is also investigated. In particular, the elimination of X-outliers followed by the one of Model-outliers enables to divide MAE and $RMSE$ (Root Mean Square Error) by 2 and 3 respectively, while reducing overfitting. The elimination of y-outliers does not display a significant effect on the model per-

formance. During model construction and deployment, the AD serves to verify the position of the test data and of different categories of molecules with respect to the training data and associate this position with their prediction accuracy. For the data that are found to be close to the training data, according to *RF confidence*, and display high prediction errors, tSNE 2D representations are deployed to identify the possible sources of these errors (e.g., representation of the chemical information in the training data). More generally, the results show that anomaly scores calculated by the *RF confidence* method are not sufficient to judge the reliability of a prediction for a new molecule. Indeed, some new molecules with low/high anomaly scores were poorly/well predicted.

As for perspectives, further research needs to be conducted to identify more appropriate anomaly score metrics to judge the reliability of a prediction for a new molecule. For example, a combination of different anomaly score metrics can be envisioned to benefit from the information contained in each metric. Besides, rules to define the borders of the AD also require further study. Another way of improvement would be to further reduce the dimensionality of the models and try to better understand how the investigated AD methods behave in lower dimension before switching to higher dimension. Furthermore, it would be interesting to include extrapolated test sets during the model construction phase to better understand how the model generalizes in case of extrapolation. Finally, the AD methods used in this work were implemented with Python's default parameters and therefore the effect of these parameters could be further studied (e.g., perplexity in tSNE representations).

3.2 Outline of the publication

1. **Introduction**
2. **Overview of the methods for AD definition/outlier detection in descriptor space and their compatibility with high-dimensional data**
   - 2.1 Classical approaches
   - 2.2 Approaches for high-dimensional data
3. **Data set and methods**
   - 3.1 Data set and ML/QSPR models
   - 3.2 AD visualization
   - 3.3 AD definition as a data preprocessing method (substudy 1)
   - 3.4 AD definition during ML model construction (substudy 2)
   - 3.5 AD definition during ML model deployment (substudy 3)
4. **Results**
   - 4.1 AD definition as a data preprocessing method (substudy 1)
     - 4.1.1 Outlier detection
     - 4.1.2 Effects of X-outliers on the model performance
     - 4.1.3 Effects of Model-outliers on the model performance
     - 4.1.4 Effects of X-outliers and Model-outliers on the model performance

3.3 Publication *"On the Development of Descriptor-Based Machine Learning Models for Thermodynamic Properties: Part 2 - Applicability Domain and Outliers.", published on 18 December 2023*

algorithms

MDPI

Article

On the Development of Descriptor-Based Machine Learning Models for Thermodynamic Properties: Part 2—Applicability Domain and Outliers

Cindy Trinh, Silvia Lasala, Olivier Herbinet and Dimitrios Meimaroglou *

Université de Lorraine, CNRS, LRGP, F-54001 Nancy, France; cindy.trinh.ct@outlook.com or cindy.trinh@univ-lorraine.fr (C.T.); silvia.lasala@univ-lorraine.fr (S.L.); olivier.herbinet@univ-lorraine.fr (O.H.)
* Correspondence: dimitrios.meimaroglou@univ-lorraine.fr

Abstract: This article investigates the applicability domain (AD) of machine learning (ML) models trained on high-dimensional data, for the prediction of the ideal gas enthalpy of formation and entropy of molecules via descriptors. The AD is crucial as it describes the space of chemical characteristics in which the model can make predictions with a given reliability. This work studies the AD definition of a ML model throughout its development procedure: during data preprocessing, model construction and model deployment. Three AD definition methods, commonly used for outlier detection in high-dimensional problems, are compared: isolation forest (*iForest*), random forest prediction confidence (*RF confidence*) and k-nearest neighbors in the 2D projection of descriptor space obtained via t-distributed stochastic neighbor embedding (*tSNE2D/kNN*). These methods compute an anomaly score that can be used instead of the distance metrics of classical low-dimension AD definition methods, the latter being generally unsuitable for high-dimensional problems. Typically, in low- (high-) dimensional problems, a molecule is considered to lie within the AD if its distance from the training domain (anomaly score) is below a given threshold. During data preprocessing, the three AD definition methods are used to identify outlier molecules and the effect of their removal is investigated. A more significant improvement of model performance is observed when outliers identified with *RF confidence* are removed (e.g., for a removal of 30% of outliers, the *MAE* (Mean Absolute Error) of the test dataset is divided by 2.5, 1.6 and 1.1 for *RF confidence*, *iForest* and *tSNE2D/kNN*, respectively). While these three methods identify X-outliers, the effect of other types of outliers, namely Model-outliers and y-outliers, is also investigated. In particular, the elimination of X-outliers followed by that of Model-outliers enables us to divide *MAE* and *RMSE* (Root Mean Square Error) by 2 and 3, respectively, while reducing overfitting. The elimination of y-outliers does not display a significant effect on the model performance. During model construction and deployment, the AD serves to verify the position of the test data and of different categories of molecules with respect to the training data and associate this position with their prediction accuracy. For the data that are found to be close to the training data, according to *RF confidence*, and display high prediction errors, tSNE 2D representations are deployed to identify the possible sources of these errors (e.g., representation of the chemical information in the training data).

Keywords: machine learning; QSPR/QSAR; high-dimensional data; descriptors; thermodynamic properties; applicability domain; outlier detection

Citation: Trinh, C.; Lasala, S.; Herbinet, O.; Meimaroglou, D. On the Development of Descriptor-Based Machine Learning Models for Thermodynamic Properties: Part 2—Applicability Domain and Outliers. *Algorithms* **2023**, *16*, 573. https://doi.org/10.3390/a16120573

Academic Editors: Szymon Łukasik, Piotr A. Kowalski and Rohit Salgotra

Received: 20 October 2023
Revised: 1 December 2023
Accepted: 5 December 2023
Published: 18 December 2023

1. Introduction

The adoption of QSPR/QSAR (Quantitative Structure Property/Activity Relationship) models has become very widespread for the prediction of various properties (e.g., physico-chemical, environmental, toxicological, safety-related) or activities (e.g., biological, pharmacological) on the basis of molecular structure. Being trained with a defined database of molecules, the applicability of these models to a new molecule is, however, limited to

the so-called "applicability domain" (AD), i.e., the chemical structure and response space within which a model can make predictions with a given reliability [1]. The chemical structure refers to the structural and physico-chemical information that is provided by the descriptors, while the response corresponds to the predicted property or activity, namely the enthalpy of formation or the entropy, in the present study. This AD is crucial, as the final goal of QSPR/QSAR models is to be able to apply them for the prediction of the target endpoint of new molecules (i.e., not used during the model training and validation) and it is well known that the use of a ML model outside its training domain can be risky [2,3]. In other words, the more distant a point is from the training domain, the more unreliable the prediction. Driven by the motivation to generalize the usage of QSPR/QSAR models for regulatory purposes (e.g., REACH regulation for Registration, Evaluation, Authorization and Restriction of CHemicals), in 2004, the OECD (Organisation for Economic Co-operation and Development) defined a set of five principles for (Q)SAR validation, including the necessity to explicitly define the AD [4]. Nevertheless, this AD definition step is still being overlooked or poorly defined in many studies, as highlighted in [3,5]. However, knowing if a ML model is extrapolating or not will improve its acceptability and transparency, as is the case when investigating the explainability of a ML model. Indeed, some studies revealed a decrease in extrapolation performance with distance from the training domain, which can be more or less pronounced depending on the ML model [2,6,7]. Finally, there is also no clear overview about which method to employ to define the AD in a given problem, especially when dealing with high-dimensional data (e.g., descriptor-based QSPR/QSAR models can contain up to several tens, hundreds or thousands of descriptors). Most classical AD approaches are indeed dedicated to problems with few dimensions (i.e., less than ten), as will be discussed in Section 2.

The AD of a model is defined by the structural, physico-chemical and response information contained in the training data used to build the model [4]. Concretely, if a molecule lies outside the AD (i.e., extrapolation case, the molecule has low similarity with the molecules of training data), the prediction is more likely to be unreliable. However, even when properly defined, the AD should be considered cautiously since it does not constitute a strictly delimited space, meaning that the prediction for a molecule inside/outside the AD is not necessarily 100% reliable or false, respectively. Besides, it would be unrealistic to consider the existence of an explicitly defined frontier between reliable and unreliable predictions; one should rather approach this discussion from the viewpoint of a compromise between the size of the AD and the reliability of the predictions. For example, in the leverage method, which is a popular method for defining the AD, as will be described later, the frontier of the AD is generally fixed at a leverage value h^*, as defined in Equation (1). However, the strict definition of this unique rule-of-thumb threshold value is not exempt from concerns [8] and, after all, expresses this compromise (i.e., increasing h^* will reduce the reliability of the predictions and vice versa [4]).

A condition for a point to belong to AD in leverage method:

$$h \leq h^* = 3(p+1)/n \quad (1)$$

where h is the leverage of a considered point, n is the number of molecules in the training set and p is the number of descriptors.

The notion of AD is closely related to the notion of an "outlier" or "anomaly". Indeed, defining the AD frontier between "normal" and "abnormal" points is identical to detecting outliers. Many definitions of an outlier can be found in reported studies [9–12]; however, a common consensus is that the term "outlier" generally refers to a point of an ensemble that is "significantly different" from the remaining points of the same ensemble. Due to their equivalence, the terms "AD definition" and "outlier detection" are therefore used interchangeably in this study. Additionally, three types of outliers can be distinguished in QSPR/QSAR studies, namely X-outliers, y-outliers and outliers towards the model (Model-outliers) [13]. X-outliers and y-outliers are outliers in the descriptor space X and the response space y, respectively. As for Model-outliers, they correspond to the molecules for

which the properties are poorly predicted by the QSPR/QSAR model and can therefore be detected only after model construction. Note that a point can belong to different categories of outliers simultaneously [14]. Note also that for a new query molecule (i.e., unknown response), only X-outlier detection methods are applicable.

More generally, even if the AD has an obvious role during the deployment of QSPR/QSAR models on new molecules, its definition remains essential at the different stages of the modeling procedure in which it fulfills different functions [1]. First, it is recommended to define the AD and eliminate the outliers as early as possible in a machine learning (ML) project, as some methods (e.g., dimensionality reduction, data scaling) are very sensitive to outliers. This initial AD definition step can be viewed as a data preprocessing step that removes outliers in order to improve the performance of the models. Then, during model construction, the AD definition can serve as a way to check whether the validation and test data are indeed within the AD defined by the training data, which is necessary as ML models are generally not reliable in case of extrapolation due to their data-driven nature. Finally, during model deployment, the AD can be used to judge the reliability of the prediction made by the model for a new query molecule. In this last case, as outliers are identified within new data, the outliers can also be referred to as novelties.

This article is the second of a series of articles aiming to construct QSPR models on the basis of ML techniques, ML-QSPR, while examining the different choices that are offered to the developer along the way and analysing the effects of each one. In the first article of the series, two ML-QSPR models were developed to predict two thermodynamic properties, namely the enthalpy of formation (H) and the absolute entropy (S) for the ideal gas state of molecules at 298.15 K and 1 bar [15]. The molecular descriptors were employed as features of the models to describe the structural and atomic characteristics of the considered molecules of the DIPPR (Design Institute for Physical Properties) database. Within an objective of discovering new chemicals for various applications, a wide range of chemical structures were considered in the training data as means to increase the AD of the developed models. The retained models were two Lasso (Least Absolute Shrinkage and Selection Operator) linear models with 100 descriptors, selected from an initial ensemble of approximately 2500 descriptors through a feature selection step based on a genetic algorithm (GA).

While the first article focused on the development of the QSPR approach from data collection to model construction, this second article aims at defining the AD of the developed models, characterized by their high dimensionality. In particular, three substudies are implemented, each corresponding to the AD definition at different stages of the QSPR procedure, as described previously; namely, the stages of data preprocessing, model construction and model deployment. Therefore, in the first part of this study (referred to here as "substudy"), the AD definition is used as a data preprocessing method, in addition to the methods implemented in the first article, to remove outliers. At this stage, the number of descriptors is particularly high, with about 2500 descriptors. Several outlier detection methods are compared and the influence of the elimination of the identified outliers on the performance of the ML models is evaluated. All the types of outliers (i.e., X-outliers, y-outliers and Model-outliers) are considered. While y-outliers and Model-outliers are detected via boxplots (interquartile range method or IQR) and standardized residuals, respectively, X-outliers are identified on the basis of three simple methods having promising compatibility with high-dimensional problems. These methods are isolation forest (*iForest*), random forest-based confidence estimation (*RF confidence*) and k-nearest neighbors in a 2D projection of the descriptor space obtained via t-Distributed Stochastic Neighbor Embedding (*tSNE2D/kNN*).

In the second substudy, the most appropriate AD definition method, as identified within the first substudy, is employed to visualize the location of the test data with respect to the AD defined by the training data. Finally, in the last substudy, the goal is to attempt a concrete deployment of the developed ML-QSPR model to new species, including the prediction and the analysis of the reliability of the predictions in terms of the AD, still based

on the AD definition method identified in the first substudy. In the last two substudies, as they occur after dimensionality reduction, they rely on fewer descriptors, namely the 100 descriptors selected by means of a GA applied on the preprocessed data without outliers. Additionally, for these same substudies, further analyses are also provided regarding the potential sources of high prediction errors.

2. Overview of the Methods for AD Definition/Outlier Detection in Descriptor-Space and Their Compatibility with High-Dimensional Data

In this section, an overview of the methods for AD definition (or outlier detection) is provided, with a focus on X-outliers. The latter are effectively the most challenging to identify in view of their high dimensionality. First, the methods that are commonly employed to define the AD of QSPR/QSAR models are presented. These methods can be classified in different categories, including chemical-based methods, range-based methods, geometrical methods, distance-based methods and probability distribution-based methods, as illustrated in Figure 1. As these have already been exhaustively reported and described in several works [1,16–19], only a quick overview is provided here, for reasons of completeness, placing special emphasis on the low compatibility of these methods with high-dimensional data. Besides, they are generally applied during model construction or deployment. Inversely, other methods for outlier detection in high dimension, presented secondly here, including confidence estimation-based methods, subspace-based methods and other specific methods (e.g., *iForest*), are more suitable to the characteristics of the three substudies investigated in this work.

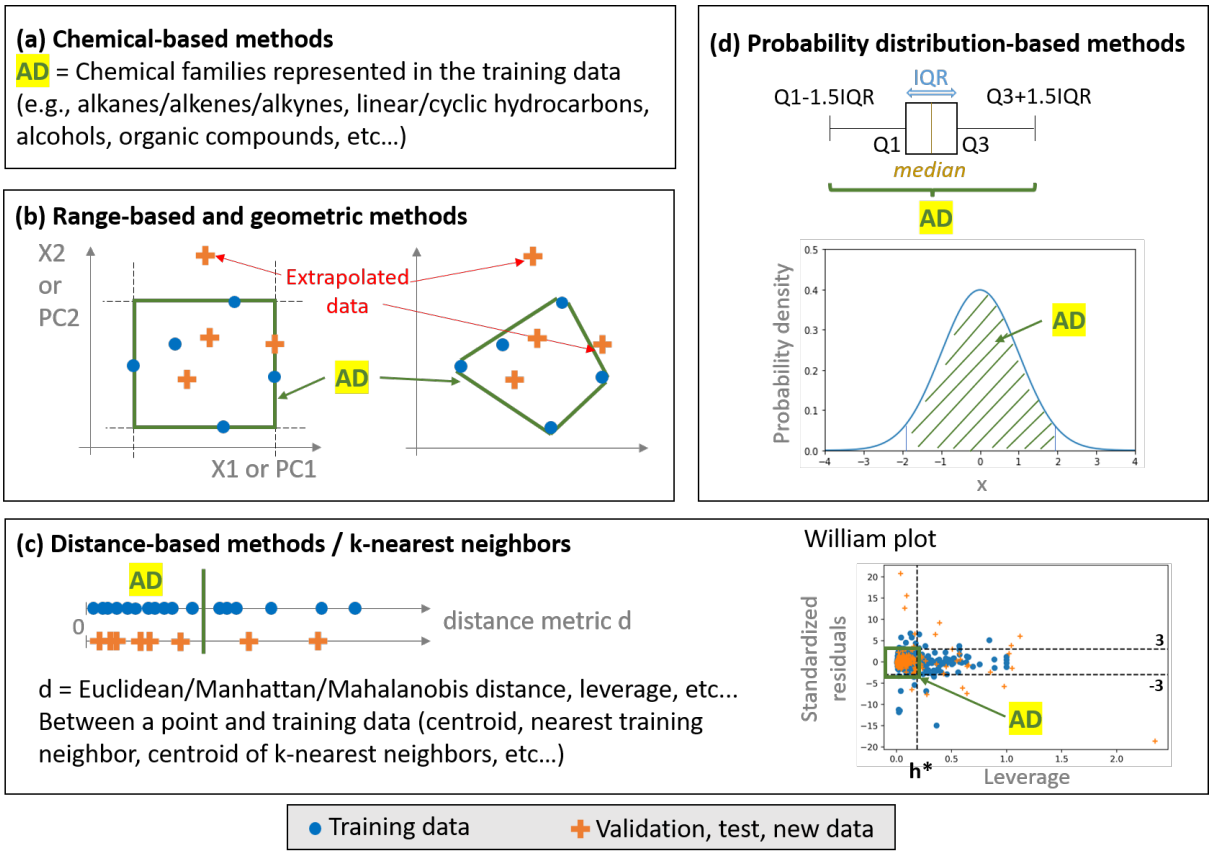


Figure 1. Classical methods for AD definition: (**a**) chemical-based methods, (**b**) range-based and geometric methods, (**c**) distance-based methods/k-nearest neighbors, (**d**) probability distribution-based methods.

2.1. Classical Approaches

A typical approach that is commonly employed in QSPR/QSAR studies involving different molecules is to consider that the AD corresponds to the chemical families on which the model was trained [20–23]. This approach, otherwise known as "chemical-based", relies on the premise that the predicted property will explicitly depend on the molecular characteristics that are used to define the chemical families (e.g., type of bonds, presence of rings or functional groups, etc.), a dependence that is not always guaranteed a priori.

"Range-based" methods use the range of each descriptor within the training data to define the AD as a hyper-rectangle of dimension equal to the number of descriptors [1]. Despite their simplicity, the major problem of range-based methods lies in the presence of empty spaces inside the hyper-rectangle making the AD description quite rough. This is even more pronounced for high-dimensional data, where points tend to become equidistant as the number of dimensions increases [9]. Empty spaces do not contain any training data and can be particularly wide in the presence of outliers. This could therefore call into question the reliability of future predictions in these regions, even though they are part of the AD. To limit the extent of the empty spaces caused by the high dimensionality, range-based methods can also be applied to descriptor-derived representations of lower dimension, such as the principal components (PC), issued by the prior implementation of a Principal Component Analysis (PCA). This combination also addresses eventual collinearities within the descriptor space.

In "geometric methods", the AD corresponds to a convex hull, as this is the smallest convex area that contains the entire training set. In a 2D space, the convex hull can be easily understood via the rubber band analogy: the rubber band tautly surrounding the training set descriptors is the convex hull. Similarly as for range-based methods, empty spaces are not detected. What is more, the calculation of the convex hull becomes computationally complex in high dimensions. In fact, as higher dimensions lead to a higher extrapolation probability [24], both geometric and range-based methods will very likely consider any new point as being outside the AD (i.e., the new point will most probably be extrapolating in at least one dimension).

"Distance-based methods" are among the most commonly used. They rely on the calculation of a distance-based metric between a given point and the training set. If the distance is below a defined threshold, the point is considered to be part of the AD and vice versa. Among popular distance metrics, one finds the Euclidean and Mahalanobis distances. The latter is popular for the detection of outliers in multivariate data, as is the case with the descriptor space [25]. Contrary to the Euclidean distance, it can identify correlations between features by means of the covariance matrix, similarly as in PCA. Graphically, the AD shape in the case of Mahalanobis distance will be more extended in the directions containing the highest variance (ellipse in 2D), and this will avoid the deletion of points that are not real outliers, as in the case of Euclidean distance (circle in 2D) [26].

Another approach that is popular in recent works is the so-called "leverage method" [1,16,27–32]. In these works, the number of features is variable, ranging from less than 10 (principally) to several hundreds or thousands (less frequently). In analogy to the Mahalanobis distance [33], the leverage quantifies the distance between a given point and the centroid of the training data X, in terms of its feature values, and provides an estimation of the potential influence that the observed response will have on its predicted value. Mathematically, the leverage values of the training samples X can be obtained via the diagonal elements h_{ii} of the hat matrix H, as defined by Equations (2) and (3). For a new query sample x_{new}, the leverage values are calculated via a similar formula, defined by Equation (4). From these equations, it is possible to identify that the major problem in the leverage method will be related to the inversion of the matrix $X^T X$. This inversion becomes problematic when the descriptor matrix X contains redundant or correlated information, or when the number of descriptors exceeds the number of samples [26]. Besides, the calculation of the matrix $X^T X$ is sensitive to outliers, as highlighted in [25,34].

Hat matrix H and leverage values h_{ii}, h_{new}:

$$H = X(X^T X)^{-1} X^T \tag{2}$$

$$h_{ii} = x_i^T (X^T X)^{-1} x_i \tag{3}$$

$$h_{new} = x_{new}^T (X^T X)^{-1} x_{new} \tag{4}$$

where X is the descriptor matrix based on training data (with n rows/observations and p columns/descriptors), x_i is the column vector of a molecule in the training data and x_{new} is the column vector of any new molecule.

In practice, leverage values (X-outliers) are plotted against standardized residuals (Model-outliers) on a Williams plot, with their corresponding thresholds. Standardized residuals are defined by Equation (5). Generally, a point belongs to the AD if its leverage value is below $h^* = 3(p+1)/n$ and if its standardized residual is between -3 and 3 [35,36]. These user-defined rule-of-thumb thresholds can be understood as follows. For the leverage threshold, the idea is that one compares the value of the leverage of a single point with the mean leverage value of all points, $(p+1)/n$. As for standardized residuals, the cutoff value of ± 3 covers 99% of the assumed normally distributed standardized residuals. Williams plots are more a graphical way to verify whether a developed model is statistically acceptable during model construction [29,36]. However, it is only possible to check if a new query molecule is an X-outlier, and not if it is a Model-outlier, as the calculation of the standardized residual is impossible (i.e., the observed property value is unknown).

Standardized residuals:

$$r_k = \frac{y_k^{observed} - y_k^{model}}{\sqrt{Var(r_{train})}} \tag{5}$$

where $y_k^{observed}$ and y_k^{model} are, respectively, the observed and the predicted responses for a molecule k and $Var(r_{train})$ is the variance of residuals of the training data.

Other distance-based methods also include techniques that utilize the k-Nearest Neighbors (kNN) algorithm. In this case, it is the distance to the nearest training point or the average distance to the k-nearest training neighbors that can be used as a means to decide whether a point lies inside or outside the AD [16]. Similar to range-based and geometric methods, empty spaces remain undetected with distance-based methods. Most importantly, all these methods suffer from the consequences of the curse of dimensionality. As the number of descriptors increases, points become equidistant whatever the chosen distance metrics. Despite several attempts to develop distance metrics that are more compatible with high-dimensional data [11], it is far more common to use dimensionality reduction methods instead. Finally, another way to improve the AD definition when using distance-based methods is to calculate the distance metrics by considering different weights for the descriptors according to their influence (e.g., coefficients in linear regression) [1].

A different class of AD definition methods is based on the use of probability density distributions. Contrary to all previous methods, this is the only class of methods that can identify empty zones inside a convex hull. Indeed, the probability density function of the training data is first estimated via a parametric or a non-parametric method. Whenever possible, the latter is generally preferred, as it does not make any assumption about the shape of the function and learns it from the data. Then, the smallest region containing a given amount of the total probability is identified as the AD. Probability density distribution-based methods can be applied to multivariate data, but their application is often limited to three dimensions. For higher-dimensional problems, these methods become computationally expensive and some assumptions are generally required. To consider both feature and response spaces, it is also possible to use joint probability distributions to define the AD. In lower dimensions, if all features are normally distributed, a popular method consists of using boxplots (IQR method), in which the points outside the AD are those located be-

low/above the lower/upper thresholds. These thresholds are, respectively, $Q1 - 1.5 \times IQR$ and $Q3 + 1.5 \times IQR$, where $Q1$ and $Q3$ are the first and third quartiles, respectively, and $IQR = Q3 - Q1$ is the interquartile range. Once again, in higher-dimensional problems, it becomes more probable that any new molecule extrapolates for at least one descriptor, similar to the range-based and geometric methods. In [19], an approach based on the Z-score of the descriptors was proposed with an additional criterion to deal with this problem in high dimension. Despite its simplicity, this approach is also applicable strictly in cases where the descriptors are normally distributed.

2.2. Approaches for High-Dimensional Data

To overcome the limitations of the previously described approaches, when it comes to high-dimensionality problems, ML methods including an estimation of the confidence of the predictions in their output (e.g., Gaussian Processes, ensemble models like Random Forest) can be employed to detect outliers. For example, [37] used an embedded AD approach, based on the probabilities assigned to the output classes of a neural network classifier, as an effective approach to analyze the likelihood of correct predictions. In other words, if the likelihood of correct prediction is low for a molecule, then the latter is likely to be an outlier. Technically, a softmax layer was applied to the network outputs to obtain the probability of each class for each molecule, where the feature space was composed of 1500 to 3700 descriptors approximately. Similarly, in [38], the predictions made by an ensemble of classification decision trees were used to evaluate the confidence of the final prediction for a molecule, and therefore the outlier character of that molecule. More concretely, the fraction of trees that voted for the majority vote was used as the confidence (or probability) of the prediction. The method was applied to problems dealing with a feature space of approximately 100 to 1000 dimensions and showed significant robustness towards the high dimensionality, in comparison to the classical Euclidean distance-based approach. The same methodology can also be implemented via an ensemble of regression trees (or of other regression models), by using the variance of the predictions as a measure of the reliability of the final prediction (i.e., the average of the predictions of the ensemble of trees) [39,40].

Gaussian Processes (GP) can also be exploited for confidence estimation. In fact, GP defines a class of ML models that are based on the prediction of a posterior distribution of functions, characterized by their mean and variance [41]. This variance can therefore be used as a measure of the reliability of a prediction. In [39], an ensemble of regression trees (RF) and GP models outperformed classical approaches in the definition of the AD. Similarly, in [42], the AD definition via confidence estimation showed better results than the other investigated methods. All these techniques are based on the confidence estimation of ML predictions and on a less binary (i.e., inside/outside AD) approach to dealing with the definition of an AD, in comparison to classical AD methods, by providing a more shaded evaluation of the prediction reliability according to different levels of confidence. This characteristic renders them more compatible with high-dimensional problems.

Among the different outlier detection techniques that can be employed in high-dimensional problems, the methods based on projections in lower-dimensional subspaces are also very popular. In [9], a point was considered an outlier if there was a lower-dimensional subspace where this point would be located in a region of abnormally low density. To avoid an exponential search of relevant subspaces, evolutionary algorithms were implemented. Additionally, the density was obtained by dividing the space into grids and then calculating a density coefficient (the sparsity coefficient) for each cell of the grids. Other works used different projection techniques, such as PCA, axis parallel subspace spanned by a point's neighbors or Hilbert space-filling curve, generally followed by a classical distance- or density-based technique, such as kernel density estimation or distance to the *kth* nearest neighbor(s), to identify outliers [43–47].

More specific methods, without projection to a subspace of lower dimension, can also be found in the literature. For example, [48] exploited angles as a measure for detecting outliers, instead of distances, as the latter are very sensitive to high-dimensional data.

Concretely, at a first step of this approach, the variance in the angles between the difference vectors of one point to the rest is calculated. If the variance is small, the analyzed point is considered an outlier, as most of the other points are located on one side (or in the same direction) of the analyzed point. In [49], an isolation-based anomaly detection method, called "isolation forest" (*iForest*), was developed to detect anomalies without using any distance or density measure, hence its compatibility with high-dimensional problems. The principle is simple, as it is based on random forest (RF). In RF, each tree splits the whole data (root node) into smaller and smaller groups until reaching leaf nodes, on the basis of decision rules at internal nodes. In *iForest*, the number of random splits to isolate a point is used to determine whether this point is an outlier or not. If the number of splits is low, the point is considered an outlier and vice versa. Indeed, it is easier to isolate a point that is located in a sparse region than another one that is part of a dense cloud of points. Similar to RF, an ensemble of trees is built to obtain an average number of splits to isolate each point. Another advantage of this method is that it can be readily implemented via the Scikit-Learn library in Python. Additional examples on both subspace-based and full-space-based methods for outlier detection in high dimensions can be found in [11,50,51], while further discussion on the effects of high dimensionality is available in [50,52,53].

3. Dataset and Methods

3.1. Dataset and ML/QSPR Models

In the previous article, two ML-QSPR models were built for the ideal gas phase standard enthalpy of formation and entropy, based on data from the DIPPR database [15]. These properties will be denoted as enthalpy (H) and entropy (S) in the rest of this article, for simplicity. Each molecule was represented by a vector of 5666 descriptors calculated using the software AlvaDesc from Alvascience [54,55]. After a series of preprocessing steps, the dataset was finally composed of 1785 molecules $\times$ 2506 descriptors for the enthalpy and of 1747 molecules $\times$ 2479 descriptors for the entropy. This dataset will be referred to as the "preprocessed data" in the rest of this article. More precisely, data preprocessing was composed of the following steps: elimination of the missing values, elimination of quasi-constant descriptors and elimination of correlations among descriptors. Finally, in this previous article, the number of descriptors was further reduced to 100 via a GA-based dimensionality reduction step to improve the interpretability of the models, in comparison to the initial 2506 or 2479 descriptors (for enthalpy and entropy, respectively). Further reduction of the number of descriptors is possible, but at the cost of prediction accuracy. A large diversity of molecules was also considered to broaden the AD. The best performing models turned out to be Lasso (Least Absolute Shrinkage and Selection Operator) models, based on the results averaged over five different train/test splits and in comparison with the different screened ML models. At last, a hyperparameter optimization step was also performed; however, this will not be considered in this article, as the main focus here is understanding the AD definition (or outlier detection) in a high-dimensional problem.

3.2. AD Visualization

The AD corresponds to the chemical structure and response space within which a model can make predictions with a given reliability. In this work, the AD will be graphically represented according to the template shown in Figure 2, and referred to as "AD plot". The representation is highly inspired by William plots, but by replacing the leverage on the x-axis with another distance (or X-outlier) metric, it becomes more compatible with high-dimensional data, as will be presented in the next section. This x-axis refers to the "chemical structure space" of the AD definition given previously.

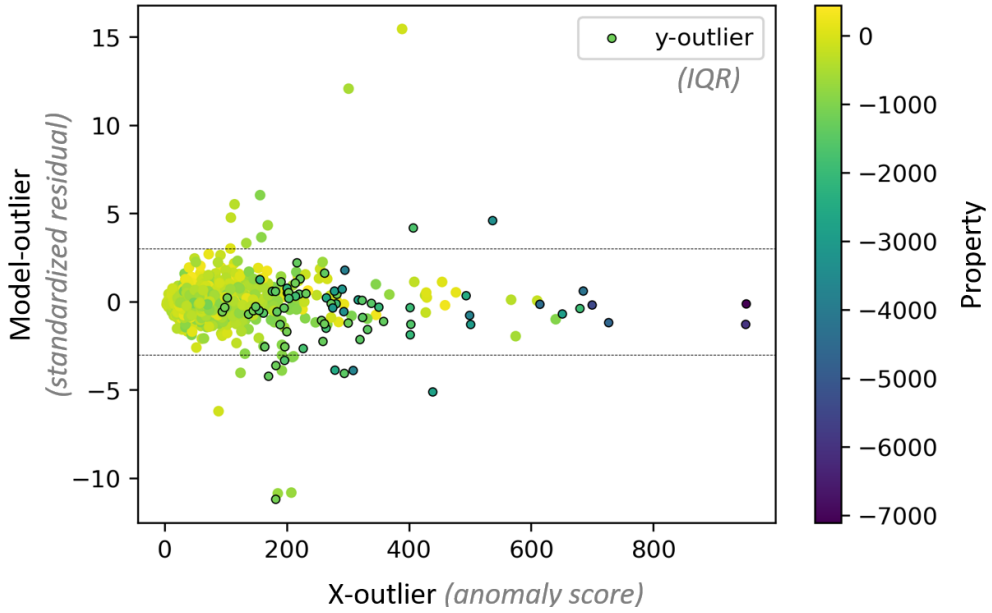


Figure 2. AD plot employed in this study. The anomaly score on the x-axis is computed via different methods: *iForest*, *RF confidence* or *tSNE2D/kNN*.

On the y-axis, the same standardized residuals as those described in William plots are considered. The combination of x- and y- axis permits us to visualize the chemical structure space within which a model can make predictions with a given reliability. In other words, it shows the success of prediction of the molecule structures considered as being part of the AD. In case of poor prediction, the concerned molecules can be considered as Model-outliers.

Additionally, a color bar can be used to visualize the property values of DIPPR data, referring to the "response space" in the AD definition and highlighting eventual y-outliers (the points with dark circles in Figure 2 are y-outliers according to the IQR method). The response space is not represented/investigated in the leverage method. Indeed, it is not possible to know if the property value of a new molecule is included in the range of property values of the training data. Besides, the extrapolation capacity of a ML model normally concerns the X-space and not the y-space. In the case of QSPR/QSAR studies, which are based on the similarity principle (i.e., similar structures lead to similar response values), X- and y- spaces are, however, somehow related. As part of this study, y-outliers will be briefly investigated.

While in the leverage method, the AD borders are defined according to widely employed rules-of-thumb, as described earlier, the borders in the case of the investigated AD methods for high-dimensional problems will be further evaluated in this work. Nevertheless, in the following section, all AD plots display the lines $y = -3$ and $y = 3$, similarly as in William plots for reasons of legibility.

3.3. AD Definition as a Data Preprocessing Method (Substudy 1)

The first part of this work lies in the identification of the outliers in the preprocessed data through different AD definition methods and the evaluation of how the elimination of these outliers may impact the performance of the models. Depending on the type of outliers, different methods are employed.

For X-outliers, three methods are compared: isolation forest (*iForest*), random forest-based confidence estimation (*RF confidence*) and k-nearest neighbors in a 2D projection of the descriptor space obtained via t-Distributed Stochastic Neighbor Embedding (*tSNE2D/kNN*). Each of these methods can effectively be exploited to calculate anomaly scores for all the molecules, and were identified as common methods for outlier detection in the presence of high-dimensional data in Section 2.2. Then, the molecules with an anomaly score above

a given threshold are eliminated. For *iForest*, it is represented by an anomaly score [49], which is negatively correlated to the average number of splits necessary to isolate a given point: the higher the anomaly score, the lower the number of splits, and the more isolated and the more abnormal the point of interest. As for *RF confidence*, the standard deviation of the individual trees' predictions serves as a judgement basis of the reliability of a prediction provided by the ensemble of trees: the higher the standard deviation, the less reliable and the more abnormal the point. Indeed, a high standard deviation means that the individual trees are associated with very different predictions for the same sample, i.e., they are not able to "agree" on a reliable final prediction because the sample is too complex (e.g., specific/outlier behavior of the sample). For *tSNE2D/kNN*, the anomaly metric used is the average Euclidean distance of a point to its three nearest neighbors. The higher the distance, the more abnormal the point considered. In particular, this Euclidean distance is computed in the tSNE 2D space, tSNE being a nonlinear dimensionality reduction method principally used for data visualization in a 2D or 3D space [56]. Finally, note that all three X-outlier detection methods were applied with Scikit-Learn default parameters.

Concerning y-outliers and Model-outliers, they were identified via the IQR method and standardized residuals, respectively (see Equation (5)). In the case of the IQR method, y-outliers are the points located outside the thresholds of $Q1 - 1.5 \times IQR$ and $Q3 + 1.5 \times IQR$, as explained previously. For Model-outliers, the molecules with absolute standardized residuals above a given threshold were considered as outliers. The residuals were obtained via a Lasso model trained on the whole preprocessed data. Note that the molecules identified as Model-outliers here are not strictly Model-outliers as defined in the introduction. Indeed, the latter are identifiable only after model construction (i.e., after eventual dimensionality reduction followed by model training and validation), while here we are still at the preprocessing stage. This can be viewed as a preventive treatment, since a late detection of Model-outliers would lead to a necessity to repeat dimensionality reduction and model construction steps after removal of the "real" Model-outliers.

To evaluate the impact of the outliers on the performance of the models, different thresholds were implemented to eliminate the outliers according to the aforementioned detection methods. Then, the resulting preprocessed data without outliers were split into training and test sets, the ratio between them being fixed at 80:20, and scaled via a standard scaling method. This ratio for data splitting and this scaling method were indeed identified as well performing in the first article of the series [15]. To better integrate the effect of data splitting on the performance of the models, five different training/test splits were considered. These five splits were defined so that each molecule belongs to the test set exactly once among the different splits. Note that outlier elimination was performed on preprocessed data and before data training/test splitting in order to keep the training/test ratio intact. Finally, the Lasso models were trained and validated for each of these splits, without performing dimensionality reduction. In the results section, the presented model performances are averaged over these splits, and the error bars displayed on the figures are the corresponding standard deviations. The considered performance metric is the mean absolute error (MAE).

In particular, four configurations are implemented and compared, as summarized in Table 1. In the first configuration, the effect of X-outliers is investigated. For each X-outlier detection method, different amounts of X-outliers are eliminated from the preprocessed data, namely 0%, 10%, 30% and 50% (based on anomaly score). The subsequent preprocessed data without X-outliers, for the different elimination thresholds and X-outliers detection methods, are then submitted to data splitting, scaling and model training/test, as described in the previous paragraph.

Concerning the second configuration, the effect of Model-outliers is studied via a procedure similar to that adopted for the X-outliers. The only difference lies within the tested standardized residual thresholds, namely 1, 2, 3, 5 and 10, in absolute values. This means that molecules with absolute standardized residual value above 1, 2, 3, 5 and 10 are removed from the preprocessed data.

Table 1. Summary of the methods and thresholds tested for each configuration in substudy 1.

Configuration	Methods	Thresholds
1. Effect of X-outlier elimination	*iForest*, *RF confidence*, *tSNE2D/kNN*	Elimination of 0%, 10%, 30%, 50% of the molecules with the highest anomaly scores
2. Effect of Model-outlier elimination	Standardized residuals	Elimination of the molecules with absolute standardized residuals above 1, 2, 3, 5 and 10
3. Effect of X-outlier and Model-outlier elimination	Layouts: 1-Simultaneous elimination 2-Model-outlier then X-outlier 3-X-outlier then Model-outlier (*RF confidence* for X-outlier, Standardized residuals for Model-outlier)	*RF confidence*: 150 kJ/mol for enthalpy, 50 J/mol/K for entropy. Standardized residuals: 2 kJ/mol or J/mol/K.
4. Effect of y-outlier elimination	IQR	Elimination of the molecules with y-values outside the thresholds of $Q1 - 1.5 \times IQR$ and $Q3 + 1.5 \times IQR$

In the third configuration, the combined effect of X-outliers and Model-outliers is analyzed. In particular, the three following layouts are compared, for a given X-outlier detection method (*RF confidence*) and for given thresholds in the detection of X-outliers and Model-outliers:

- Layout 1: simultaneous elimination of X-outliers and Model-outliers.
- Layout 2: elimination of Model-outliers followed by elimination of X-outliers.
- Layout 3: elimination of X-outliers followed by elimination of Model-outliers.

Finally, for y-outliers, the thresholds are those as defined earlier using the IQR method, and the effect of their elimination on the model performance is analyzed.

3.4. AD Definition during ML Model Construction (Substudy 2)

Once the best-performing outlier detection method has been identified and the outliers eliminated (substudy 1), the obtained dataset, preprocessed and without outliers, is subjected to the previously presented data splitting and data scaling steps. Furthermore, a dimensionality reduction step is implemented to improve interpretability of the developed ML models. To achieve this, the same GA procedure as the one used in the first article is used to reduce the numbers of descriptors to a fixed number of 100 [15]. Finally, the Lasso models are trained and validated on the new dataset of reduced dimension. Note that the criterion of selection of the best-performing outlier detection method is based on compromise between high training/test performances and the lowest possible number of eliminated molecules.

The main goal of this second substudy is to visualize where the test data are located with respect to the AD defined by the training data. Therefore, the AD of each model is visualized on AD plots showing the standardized residuals (Model-outliers) of the different molecules versus their anomaly scores (X-outliers). The test data should be within or close to the AD to have a reliable model.

In addition to the above, tSNE 2D representations are provided in an attempt to identify the potential sources of high prediction errors (e.g., lack of training molecules in certain regions, presence of molecules with different structures but similar property values...). Note that the procedure of this substudy also applies to a potential hyperparameter optimization phase in order to check whether the validation data are indeed close to the training data. However, this study has not been pursued in the framework of the present article.

3.5. AD Definition during ML Model Deployment (Substudy 3)

The goal of this final substudy is to show a concrete deployment of the developed ML-QSPR models to new species (i.e., those not used for model training and validation), including the prediction and the analysis of the reliability of the predictions with respect to the defined AD. In this sense, the models resulting from substudy 2, based on the preprocessed data without outliers and with dimensionality reduction, are deployed to predict the properties of new species. These are classified into four different categories, namely Cat. A (13 species), Cat. B (5 species), Cat. C (45 species for enthalpy, 44 species for entropy) and Cat. D (12 species). Due to confidentiality issues, no further information about the chemistry of these species can be provided. The descriptor values for these new species were calculated by AlvaDesc (v2.0.8) on the basis of the optimized geometry, in MDL MOL format (containing information on atom coordinates/types and on bond types), calculated by the software Gaussian 09 (revision B.01) [57]. Calculations were performed at the CBS-QB3 level of theory [58] and conformational analysis was conducted in a systematic way to be sure to identify the structure with the lowest energy (this was performed at the B3LYP/6-31G(d) level [59]). Raw data provided by the software Gaussian were then post-processed using the GPOP software suite from Miyoshi [60] to derive the thermodynamic properties (enthalpy and entropy). The contribution of internal rotations of moieties around simple bonds was corrected by considering the hindered rotor model rather than the harmonic oscillator one when the potential energy was less than 10 kcal/mol. Note that the calculations could not be performed for all species contained in Cat. C, resulting in 21 and 39 species with available Gaussian-calculated properties in Cat. C, for H and S, respectively.

The reliability of the predictions is subsequently analyzed in relation to the distance of the new species to the AD. Again, the AD is visualized on AD plots, showing the species' standardized residuals (Model-outliers) and *RF confidence*-based anomaly scores (X-outliers). Note that, in practice, the properties of new species are unknown; hence, their standardized residuals are also unknown. In this work, the enthalpy and/or entropy of the new species were calculated by DFT on Gaussian to serve as reference values for comparison with the predictions. The availability of the standardized residuals for the new species is utilized to improve the understanding and analysis of the AD. Besides, similarly as in substudy 2, tSNE 2D representations are also provided to try to identify the potential sources of high prediction errors.

All the ML models of this work were implemented using the Scikit-learn library v1.0.2 of Python v3.9.12.

4. Results

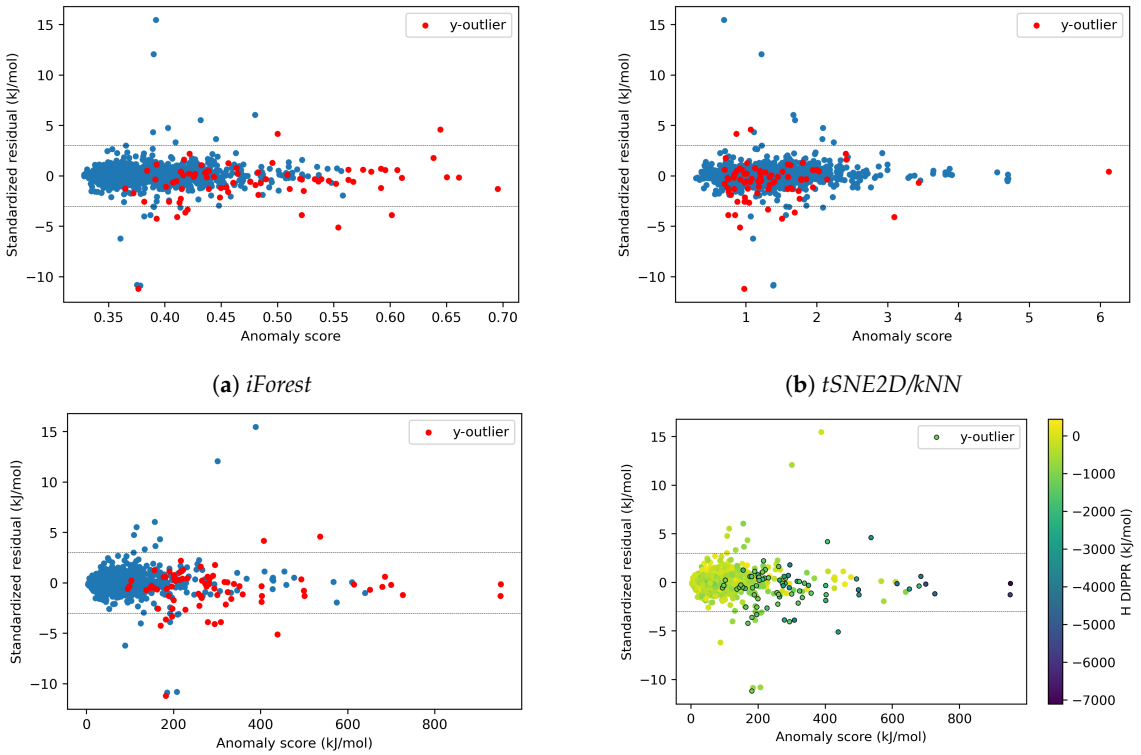
4.1. AD Definition as a Data Preprocessing Method (Substudy 1)

The analysis of AD definition methods during data preprocessing was investigated for the enthalpy QSPR model and is presented below. As for entropy, the best-performing AD definition method, identified for the enthalpy, is applied to eliminate the outliers. All the results are provided below.

4.1.1. Outlier Detection

X-outliers, y-outliers and Model-outliers are detected on the preprocessed data via the different methods presented in Section 3.3. Figure 3a–c show a visual representation of the preprocessed data for the enthalpy only, according to the different categories of outliers, as well as according to the three studied methods for X-outlier detection. In particular, the x-axis corresponds to an anomaly score specific to each X-outlier detection method, while the y-axis corresponds to the standardized residual between the reference response value (DIPPR) and the predicted one by the Lasso model trained on the whole preprocessed data. Points identified as y-outliers are colored in red. Note that the implementation of the leverage method for X-outlier detection is not presented here due to the occurrence of numerical issues in the calculation of the hat values, probably due to existing multicollinearities

between the descriptors (more than 2000 descriptors) that resulted in negative eigenvalues of the matrix $X^T X$ [26,61]. This is evidence of the limitations of the leverage method when applied to high-dimensional data. In addition, tSNE was preferred over PCA for projecting the descriptor space into a lower-dimensional space, since the number of PCs necessary to describe 95% of the data variance exceeds 200, with the first component containing only 20% of the variance (the numbers given are for the enthalpy, but the situation is similar for the entropy).



(**a**) *iForest* (**b**) *tSNE2D/kNN*

(**c**) *RF confidence* (**d**) *RF confidence* with H values

Figure 3. Comparison of three X-outlier detection methods on the preprocessed data for **enthalpy**: (**a**) *iForest*, (**b**) *tSNE2D/kNN*, (**c**) *RF confidence*, (**d**) *RF confidence* combined with H values. y-outliers are identified via the IQR method.

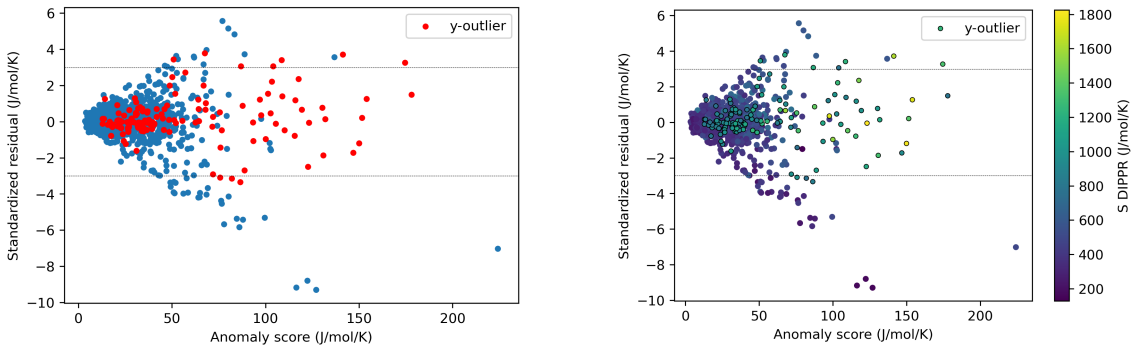
For all three X-outlier detection methods, similar behavior can be observed in Figure 3a–c. Points showing a high score in the abscissa metric (i.e., considered as the "most abnormal" in descriptor space) tend to be better predicted (i.e., with lower absolute standardized residual values) than some points with lower abscissa values. It seems as if, even though these points are located "further away" in the descriptor space (i.e., in terms of the employed metric) than the majority of data, the model succeeds in "extending" the response hypersurface to include them [33].

The molecules displaying the highest absolute standardized residuals (Model-outliers) and the highest anomaly scores (X-outliers), according to the three investigated detection methods, are shown in Tables A1 and A2, respectively. The first category (i.e., Model-outliers) mainly includes molecules containing halogen, polyfunctional and silicon compounds, while in the second category (i.e., X-outliers) it is seen that *iForest* and *RF confidence* identify some common molecules, such as large silicon or halogen compounds. All these identified families share a common element of being poorly represented, as there is a lack of molecules containing the characteristic heteroatoms of these families in the database. This could be one of the reasons for their identification as outliers; for example, they can be easily isolated (*iForest*) or display high standard deviations in the individual trees' predictions (*RF confidence*). Note also that polyfunctional compounds combine several functional groups,

thus conferring to these molecules a partial similarity, in terms of some descriptor values, to other species that also contain these groups, despite their overall different structures and enthalpy and/or entropy values. Additionally, for the two first molecules of Table A1, with chemid n°3608 and 2628, another explanation for their outlier behavior could be related to the significant difference (i.e., of 700 and 600 kJ/mol, respectively) that is observed in their enthalpy values with their counterparts with the same formula. Finally, concerning X-outlier detection, *tSNE2D/kNN* does not seem to share any molecule in common with the two other methods. This is mainly due to the completely different mechanism that is employed by this method, where the molecules identified as X-outliers are the most isolated ones from the clusters that are formed in the tSNE 2D space, in contrast to tree-ensembles that are employed by *iForest* and *RF confidence*. The effects of both X-outliers and Model-outliers on the model performance are further investigated in the following parts.

Concerning y-outliers, it is seen in Figure 3a–c that, when *RF confidence* is employed, these are mostly located in the areas with high anomaly scores, outside the cluster with a high density of points. Accordingly, their elimination can be carried out via the elimination of X-outliers in this case. Figure 3d, where the *RF confidence* plot is reproduced, coupled with a color scale indicating the enthalpy values of the molecules, seems to confirm that a high percentage of both X- and y-outliers is displayed by molecules of lower enthalpy values. This observation, in conjunction with the distribution of enthalpy values in the dataset (cf. the first article of the series [15]), corroborates the hypothesis of poor representation of these molecules in the dataset (e.g., very large molecules). In this case, the elimination of these points is questionable. In fact, points from poorly represented domains may be useful to a model whose applicability will need to include such molecules. In the case of entropy, the elimination of y-outliers via the one of X-outliers does not seem feasible, as shown in Figure 4, where a non negligible amount of y-outliers is contained in the area with low *RF confidence* anomaly scores, within the cluster with a high density of points. This could be related to the eventual existence of property cliffs, namely molecules that have similar descriptor values but very different property values, thus posing significant problems for QSPR/QSAR modeling studies, as these rely on the similarity principle [62]. The existence of such issues in the dataset is also evidence of the limitation of the descriptor-based molecular representation approach in these studies.

Finally, looking at Figures 3 and 4, it seems that, for *RF confidence*, the regions with the lowest anomaly scores seem to contain less molecules with high absolute standardized residuals, which are located in a less dense region with higher anomaly scores. This effect is particularly visible for entropy. All these elements highlight a possible relation between *RF confidence* anomaly score and high prediction errors, which could be exploited during ML model deployment to new species to assess the reliability of the predictions. In other words, the aim is to investigate whether a new species with a low *RF confidence* anomaly score is more likely to provide more reliable model predictions and the opposite.

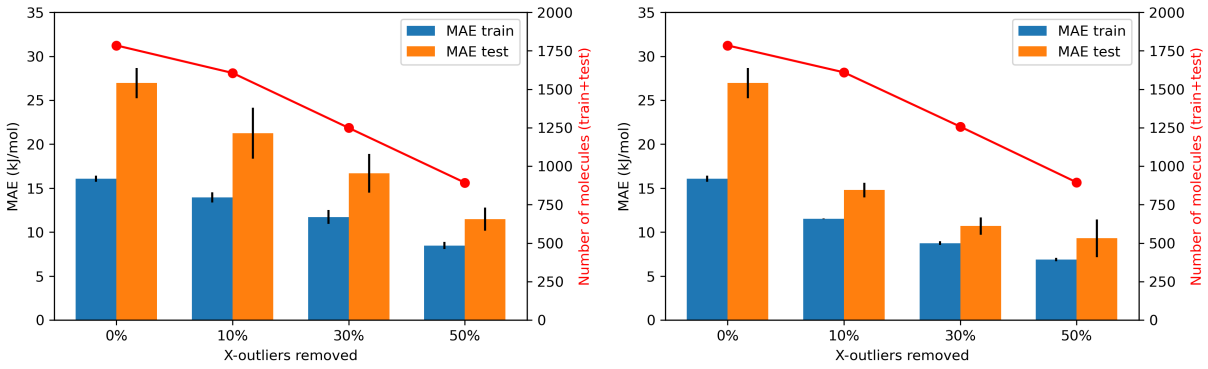


(**a**) *RF confidence* (**b**) *RF confidence* with S values

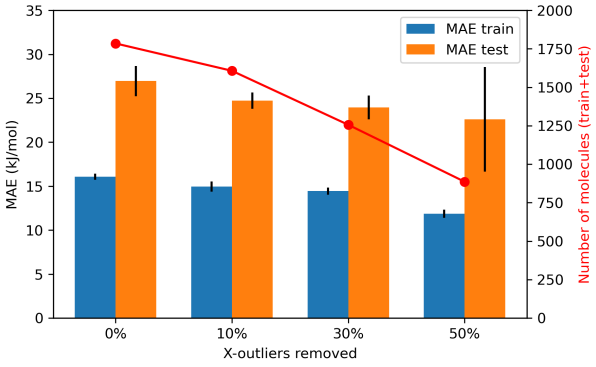
Figure 4. X-outlier detection on the preprocessed data with *RF confidence* for **entropy**.

4.1.2. Effects of X-outliers on the Model Performance

To better understand the effects of X-outliers on the performance of the models, different amounts of X-outliers (0%, 10%, 30% and 50%) are eliminated from the preprocessed data on the basis of their anomaly score (N.B. the elimination of a percentage of outliers in this case reduces to the elimination of the same percentage of all molecules, with the highest anomaly score). The results are presented for enthalpy in Figure 5a–c, respectively, for *iForest*, *RF confidence* and *tSNE2D/kNN*. In addition, in Figure 6a,b, the results of all three methods are compared for the training and test sets, respectively. Note that the same parameter initialization was used for the three methods (hence, the same *MAE* value at 0% removal). First of all, it can be clearly observed that the elimination of X-outliers with high anomaly scores improves the model performances on both the training and test datasets for *iForest* and *RF confidence* but not for *tSNE2D/kNN*. For example, in *RF confidence*, for which the most important improvement is observed, the elimination of the 10% most important outliers enables us to reduce the test *MAE* by 40%. These outliers correspond to 180 molecules approximately, for which the *RF confidence* anomaly scores are above 150 kJ/mol. A possible explanation of the good performance of *RF confidence* could be related to the simultaneous elimination of a higher percentage of y-outliers along with the X-outliers, as shown previously for enthalpy. In case of extreme elimination percentages of data points, as X-outliers (e.g., 70%, 90%), the test *MAE* will start to increase at some point as the number of remaining molecules will become too low for the model to learn the relationship between descriptors and property and therefore to generalize well. More generally, the elimination of X-outliers (when they are correctly identified with an adapted method) will improve the model's performance; however, care must be taken to ensure that they are properly distinguished from other "useful" data points. Indeed, X-outliers will affect the response surface and therefore deteriorate the predictions for the other data.

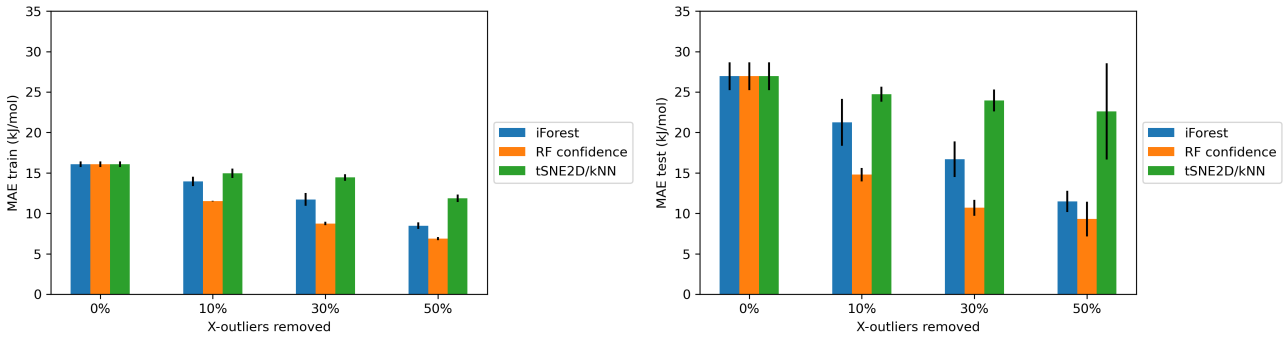


(**a**) *iForest* (**b**) *RF confidence*



(**c**) *tSNE2D/kNN*

Figure 5. Evolution of Lasso model performance in predicting the **enthalpy** under different proportions of X-outliers removed from the preprocessed data: (**a**) *iForest*, (**b**) *RF confidence*, (**c**) *tSNE2D/kNN*.

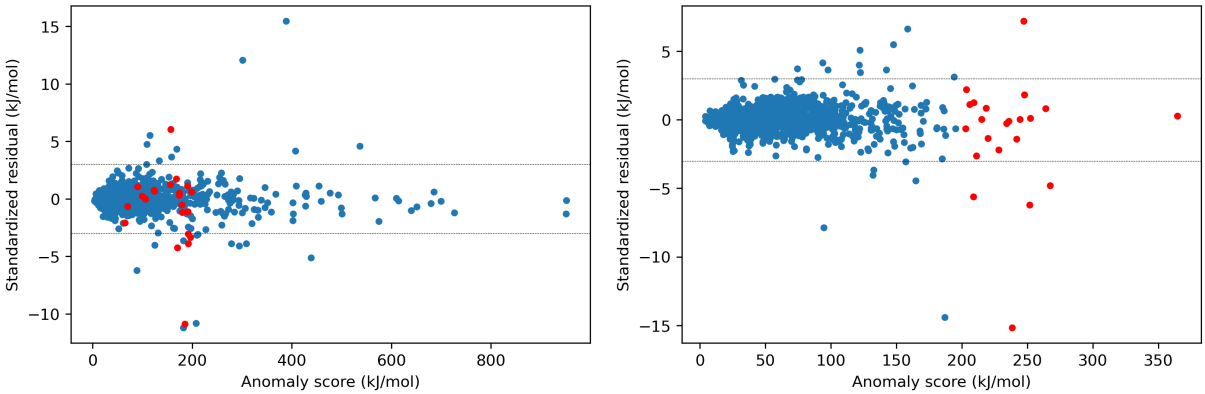


(**a**) Train data set (**b**) Test data set

Figure 6. Comparison of Lasso model performance in predicting the **enthalpy** under different proportions of X-outliers removed from the preprocessed data via the three tested methods on the (**a**) train and (**b**) test datasets.

The regression model can therefore be highly influenced by the presence of X-outliers, but their identification depends on the employed method. However, this improvement in the model performance, caused by the removal of outliers, comes with a certain cost, namely the loss of molecules from the dataset (cf. red curves in Figure 5) and, consequently, a modification of the AD of the models. Note that the removal of the X-outliers also reduces overfitting (i.e., observed as a decrease in the difference between the training and test performances), but again only for *iForest* and *RF confidence* methods. Indeed, on the one hand, if an outlier is present in the training set, the regression model will extend in order to include it, thus lowering its generalization ability. On the other hand, if an outlier is present in the test set, the trained model will most probably perform poorly in predicting its property, as it will not have been trained on similar descriptor values. Finally, the tSNE 2D representation and the selected anomaly score seem to be inefficient in identifying outliers whose elimination will lead to an improvement in the model performance. Indeed, the distance to the k-th nearest neighbors is probably not explicit enough as a metric to identify such points, given also the observed differences in the property values of neighboring points (i.e., property cliffs). Besides, the tSNE 2D representation displays some limits. These observations and the limits of some of the employed methods, such as tSNE 2D, are further discussed in Sections 4.2 and 4.3.

In the above observations and analyses, it is important to keep in mind that once outliers with anomaly scores above a given threshold are removed from the dataset, new values for the anomaly scores can be calculated based on the remaining data. These new values are not necessarily below the given threshold, meaning that points that were not outliers become outliers. For example, Figure 7b shows the new standardized residuals versus the new anomaly scores after removal of the points with initially anomaly scores above 200 kJ/mol (see Figure 7a). It can be observed that the new anomaly scores of some points (points in red in Figure 7b), display an increase in comparison with their initial anomaly score (points in red in Figure 7a). These red points belong to molecules from various chemical families, but mostly from the families of nitrogen and halogen compounds. These molecules become outliers with respect to the data after the elimination of the initial outliers, probably also because their structure becomes even less represented among the remaining data. Furthermore, the elimination of the X-outliers seems to reduce the range of the anomaly scores, and this also modifies the range of standardized residuals without necessarily improving it. This phenomenon is clear evidence of the direct dependence of the notion of outlier and the composition of the complete dataset, which renders the detection and treatment of outliers particularly complex.



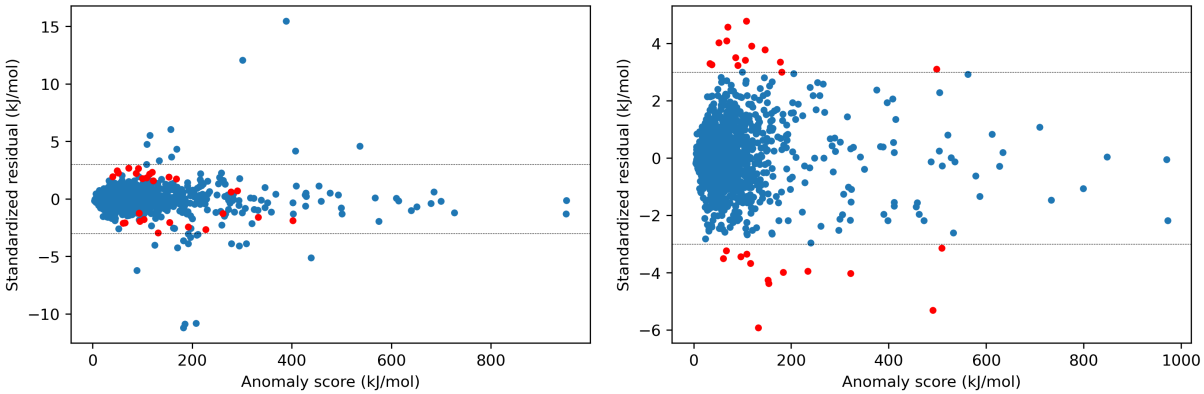
(**a**) Before outliers removal (**b**) After outliers removal

Figure 7. Comparison of the dataset (**a**) before and (**b**) after the removal of X-outliers via the *RF confidence* method for **enthalpy** (the molecules with anomaly scores above 200 kJ/mol in (**a**) are removed). In red, the molecules that were not outliers in (**a**) become outliers in (**b**).

4.1.3. Effects of Model-Outliers on the Model Performance

The same procedure as in Section 4.1.2 applies here but, this time, outliers are removed on the basis of their standardized residual values (Model-outliers) instead of the anomaly scores (X-outliers). In this sense, different thresholds of standardized residuals are tested, namely 1, 2, 3, 5 and 10, in absolute values. This means that the molecules with absolute standardized residual values above 1, 2, 3, 5 and 10 are removed from the preprocessed data.

Initially, the same effect (i.e., points that are not outliers before the removal of molecules become ones after the removal) as presented in Section 4.1.2 can be observed concerning the "data-relative" character of an outlier. Figure 8b shows the new standardized residuals versus the new *RF confidence* anomaly scores after removal of the points with initial absolute standardized residual values above three (see Figure 8a), for the enthalpy. It can be observed again that some points have increased their absolute standardized residual values after the outlier removal (points in red in Figure 8a,b). These points, corresponding mostly to molecules from the families of silicon and inorganic compounds, become less represented (at least, in a consistent manner in terms of the enthalpy values) in the new dataset, after the elimination of the Model-outliers. Besides, the elimination of the outliers with absolute standardized residuals above three seems to have reduced the range of standardized residuals, but not that of the *RF confidence* anomaly scores.



(**a**) Before outliers removal (**b**) After outliers removal

Figure 8. Comparison of the dataset (**a**) before and (**b**) after the removal of Model-outliers for **enthalpy** (elimination of the molecules with absolute standardized residuals above 3 kJ/mol in (**a**)). In red, the molecules that were not outliers in (**a**) become outliers in (**b**).

The effect of the elimination of outliers, based on their standardized residual, on the enthalpy model performance is presented in Figure 9. The x-axis represents the tested thresholds, a threshold of k corresponding to the elimination of all the molecules with absolute standardized residuals above k. Similarly as for the elimination of X-outliers based on anomaly scores, the model seems to improve its performance with the elimination of outliers with high absolute standardized residuals, while overfitting is also reduced. At the same time, the number of eliminated molecules is not as dramatic as in the case of X-outliers, allowing a wider margin of selection of the final threshold value. In fact, it is seen that only the strictest threshold of 1 kJ/mol, among the tested ones, results in a significant elimination of about 200 molecules, for a decrease in the *MAE* that remains comparable to the rest of the tested thresholds.

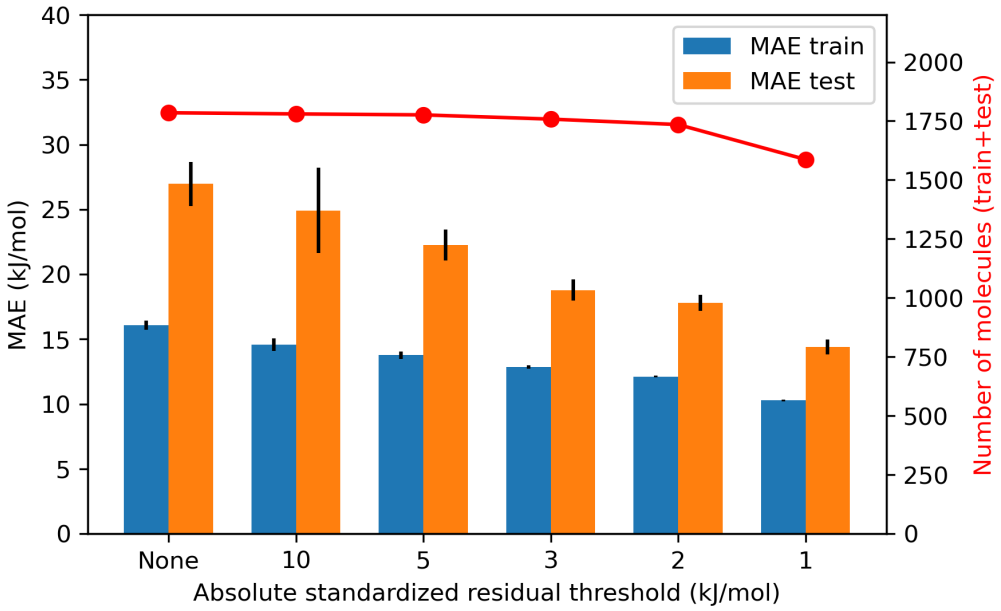
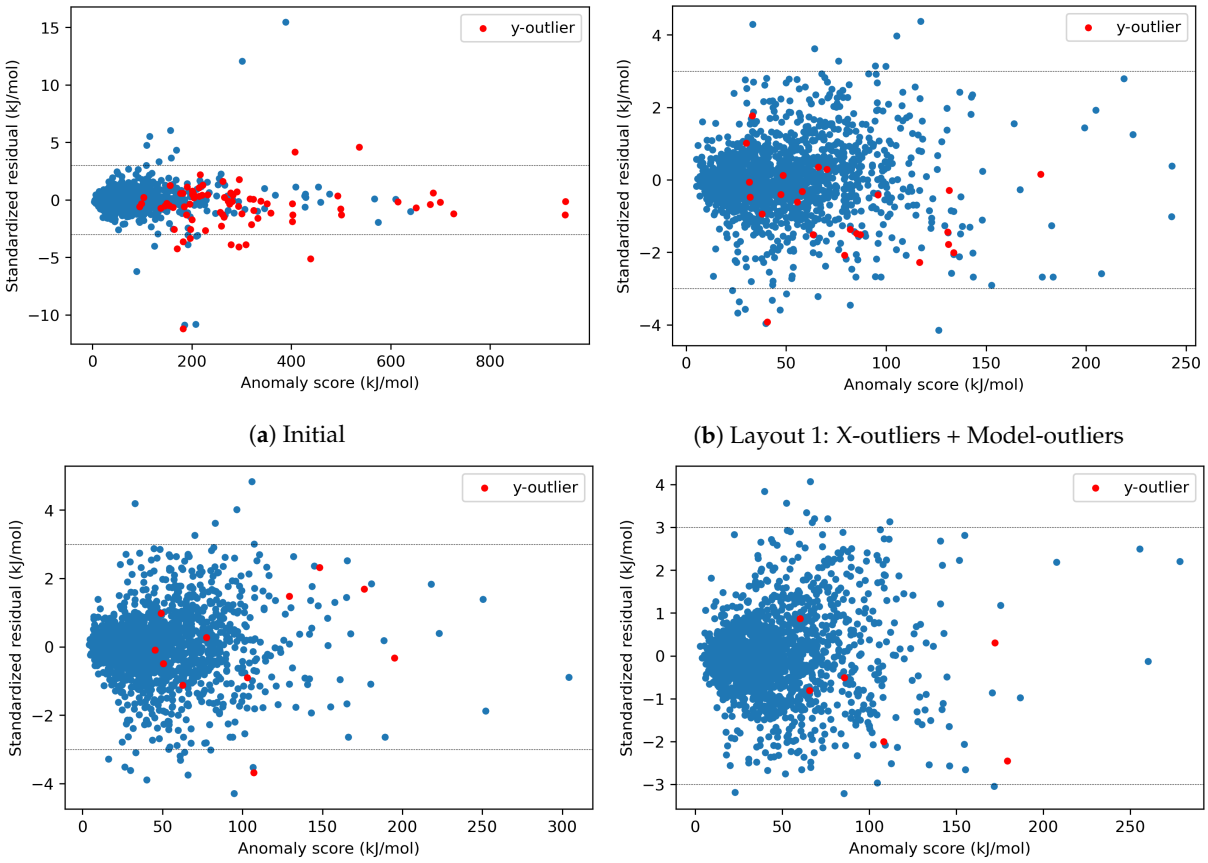


Figure 9. Evolution of Lasso model performance in predicting the **enthalpy** under different thresholds for Model-outlier elimination in the preprocessed data.

4.1.4. Effects of X-outliers and Model-outliers on the Model Performance

Previously, the effects of outliers, identified either in terms of the anomaly scores (X-outliers) or the standardized residuals (Model-outliers), on the performance of the ML models were studied individually. In this section, both aspects are considered combined, with *RF confidence* for the X-outlier detection method. More precisely, starting from the preprocessed data, the three layouts described in Section 3.3 are compared, with thresholds of 2 kJ/mol and 150 kJ/mol, respectively, for the absolute standardized residual and the *RF confidence* anomaly score.

The results for enthalpy are shown in Figure 10. In particular, Figure 10a represents the whole preprocessed dataset, prior to any outlier elimination, while Figure 10b–d, correspond to the same data but with an additional step of outlier elimination, according to Layouts 1–3, respectively. It can be observed that each outlier elimination scenario enables to obtain a cloud of points that lies inside a more restricted window, with respect to the initial dataset, even if some points that were not outliers become outliers during the elimination process. In any case, the clouds of points obtained with the three layouts look similar. Their performance and amount of molecules are also close, as observed in Figure 11.



(**a**) Initial

(**b**) Layout 1: X-outliers + Model-outliers

(**c**) Layout 2: Model-outliers > X-outliers

(**d**) Layout 3: X-outliers > Model-outliers

Figure 10. Effect of different scenarios of outlier elimination on the data set of the **enthalpy**.

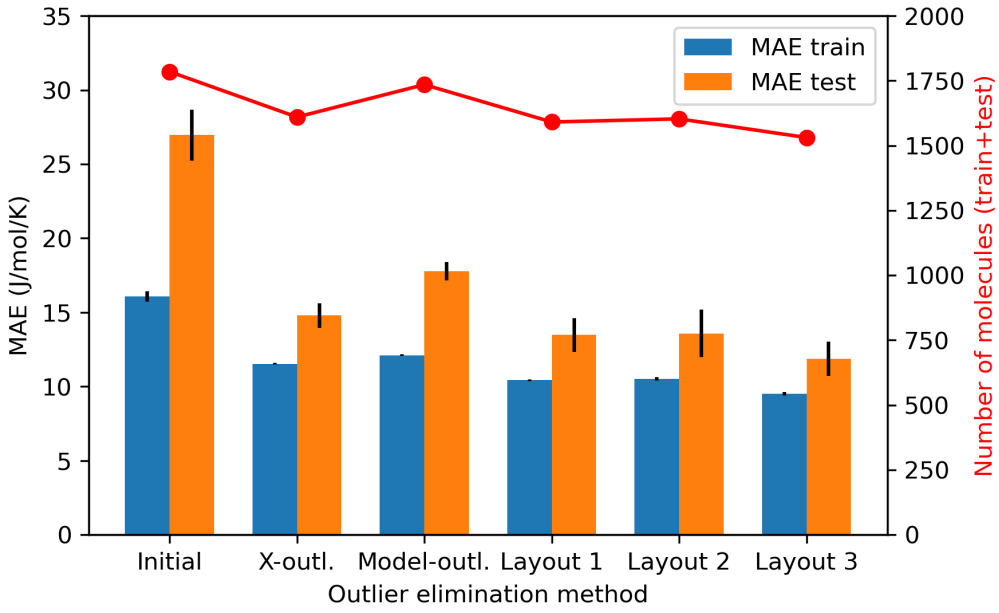


Figure 11. Effect of different scenarios of outlier elimination on Lasso model performance in predicting the **enthalpy**.

All three layouts enable us to drastically improve the performance of the model (more than when only X-outliers or Model-outliers are removed with the same thresholds), achieving an overall two-fold decrease in the test MAE. Among the three tested scenarios, a slightly better performance is obtained for Layout 3 (MAE train = 9.51 kJ/mol vs. 10.43 for Layout 1 and 10.51 for Layout 2; MAE test = 11.88 kJ/mol vs. 13.48 for Layout 1 and 13.58 for Layout 2), but this comes at the cost of a marginally increased elimination of molecules, with respect to the other two layouts (1531 remaining molecules vs. 1591 for Layout 1 and 1603 for Layout 2). Due to its higher performance, Layout 3 is therefore chosen as the outlier elimination method for the rest of this article. Note that Figure 11 highlights this compromise between the size of the AD and the reliability of the predictions, as already discussed.

Layout 3 is also applied for the elimination of outliers for the entropy, but with different threshold values for X-outliers in view of a higher density of the cloud of points in a smaller region (see Figure 4). These thresholds are set to 50 J/mol/K and 2 J/mol/K for the *RF confidence* anomaly score and the standardized residual, respectively. The resulting data are shown in Figure 12 and result again in a two-fold reduction in the model test MAE, though it is slightly better for Layout 3 (MAE train = 7.49 J/mol/K vs. 8.61 for Layout 1 and 8.8 for Layout 2; MAE test = 8.19 J/mol/K vs. 9.82 for Layout 1 and 10 for Layout 2).

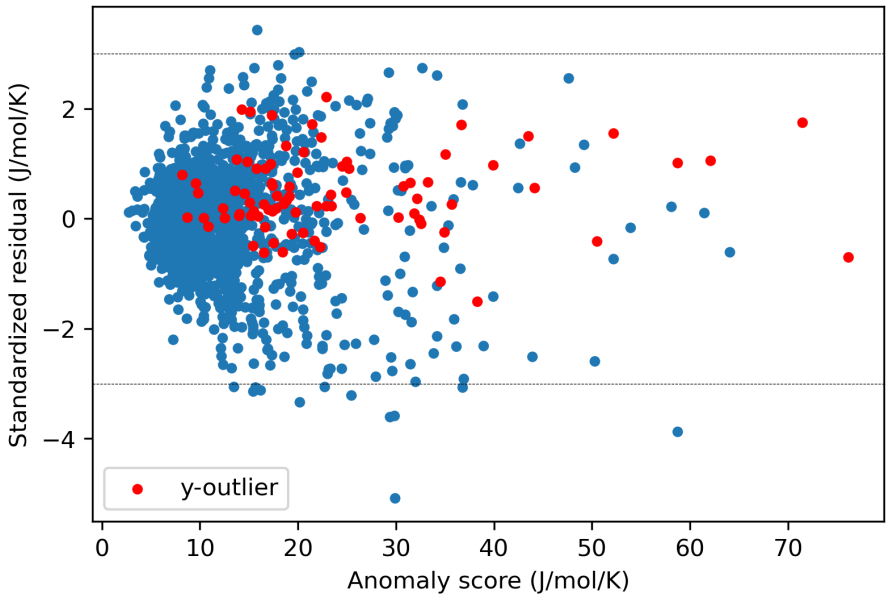


Figure 12. Data set after the elimination of outliers with Layout 3 for the **entropy**.

4.1.5. Effects of y-outliers on the Model Performance

Concerning the y-outliers still present in the dataset after the elimination of X-outliers and Model-outliers with Layout 3 (in red in Figures 10d and 12), they were not eliminated, as this did not lead to a significant improvement in the performance of the models, as shown in Table 2. In fact, the removal of y-outliers does not necessarily improve the statistical parameters, since a y-outlier is not necessarily a Model-outlier. Indeed, Figures 10d and 12 show that y-outliers have low standardized residuals. More generally, a y-outlier can have more or less impact on the overall model's performance. If a molecule is both a y-outlier and a X-outlier, its thermodynamic property can be both well and poorly predicted: it will depend on if the ML model manages to extend its response surface to include this molecule. If a molecule is a y-outlier but not a X-outlier, this molecule represents a case of property cliff (i.e., similar structures but very different property values). In this case, the response surface can be pulled toward this molecule which can deteriorate the model's performance in varying amounts. In this study, the model's performance was improved for H after y-outliers removal, but not for S (cf. Table 2). It is difficult to qualitatively explain

this difference between H and S, as the elimination of each individual y-outlier can have a positive or negative effect on the model's performance. Accordingly, the overall trend can be different in each case. Finally, further elimination of X-outliers and/or Model-outliers would have been possible, allowing us to eventually further improve the performance of the models, but to the detriment of the loss of data and a further reduction in the size of the AD. In the end, it comes down to the user to select the most appropriate methods given the problem requirements and the available dataset characteristics.

Table 2. Comparison of the performance of the models in absence and in presence of y-outliers. *MAE* and *RMSE* are in kJ/mol and J/mol/K for the **enthalpy** and the **entropy**, respectively.

Property	Outliers Elimination	Mol.	Desc.	R^2 Train	R^2 Test	*MAE* Train	*MAE* Test	*RMSE* Train	*RMSE* Test
H	Layout 3, with y-outliers	1531	2506	0.998	0.995	9.51	11.88	12.72	19.41
	Layout 3, without y-outliers	1525	2506	0.998	0.996	9.39	11.69	12.61	17.56
S	Layout 3, with y-outliers	1514	2479	0.996	0.995	7.49	8.19	9.90	11.40
	Layout 3, without y-outliers	1431	2479	0.991	0.988	7.56	8.28	10.00	11.49

Mol.: number of molecules. Desc.: number of descriptors. R^2: coefficient of determination. *MAE*: mean absolute error. *RMSE*: root mean square error.

4.2. AD Definition during ML Model Construction (Substudy 2)

4.2.1. Dimensionality Reduction on the Preprocessed Data without Outliers

To improve the interpretability of the models, a dimensionality reduction step based on GA is applied on the preprocessed data after outlier elimination through Layout 3. Note that the GA algorithm enables us to remove *descriptors* that are not relevant for the predicted property, while the elimination of X-outliers removes *molecules* that exhibit outlier behavior based on their descriptor values. The obtained performances are presented in Tables 3 and 4 for the enthalpy and entropy, respectively, where configurations with or without outliers/dimensionality reduction are compared. The removal of the outliers enables us to improve the performance of the models (configurations A vs. B, and C vs. D), although the resulting AD might be restricted due to the elimination of some molecules. The dimensionality reduction does not seem to have a major impact on the performance (configurations A vs. C, and B vs. D), despite the significant reduction in the number of descriptors that are reduced from more than 2000 to only 100, thus enhancing the subsequent interpretability of the models.

Table 3. Effects of outlier elimination (Layout 3) and dimensionality reduction (GA) on the model performance in predicting **enthalpy** (kJ/mol).

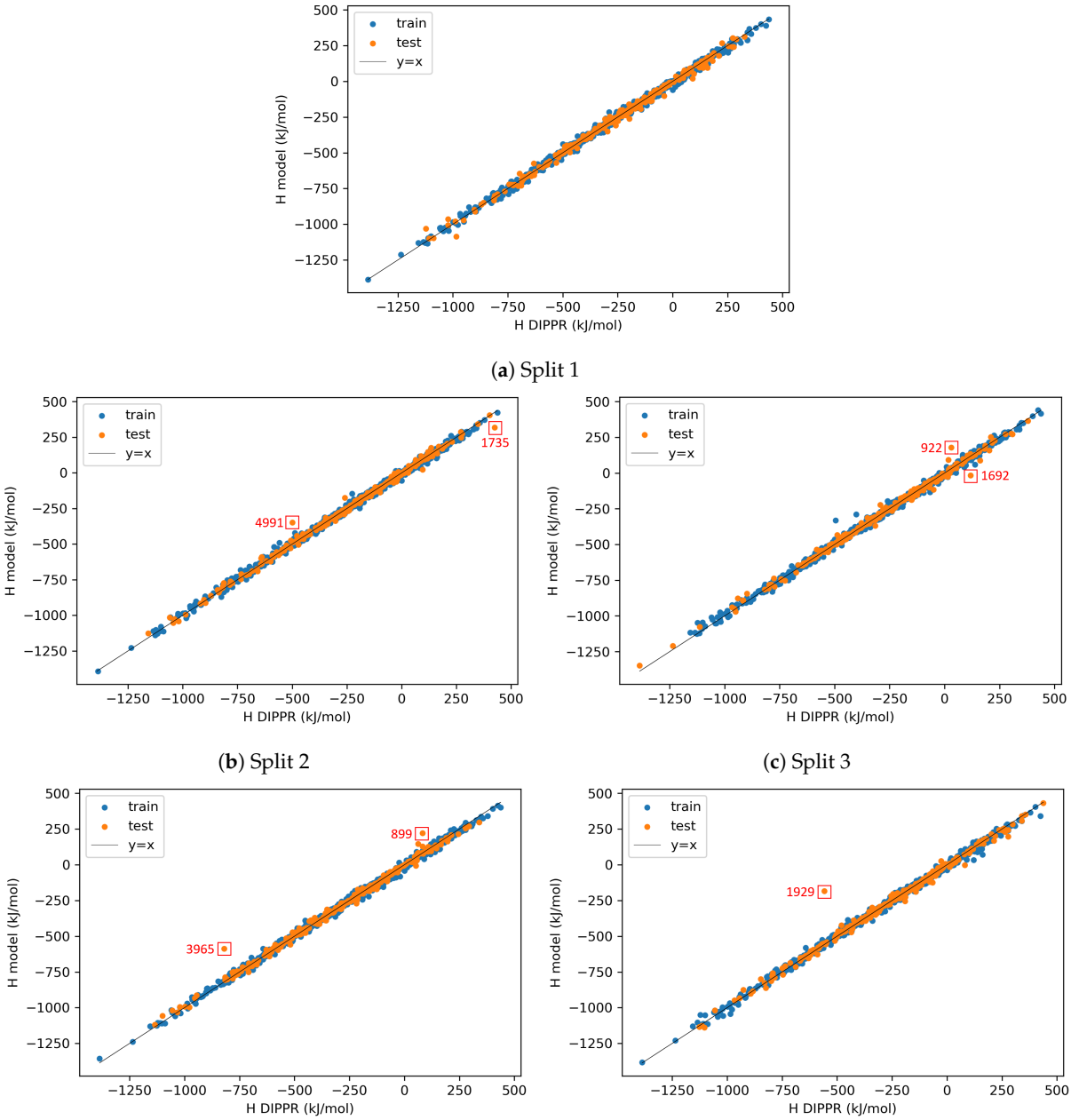
Configuration	Outlier Elimination	Dimensionality Reduction	Mol.	Desc.	R^2 Train	R^2 Test	*MAE* Train	*MAE* Test	*RMSE* Train	*RMSE* Test
A	No	No	1785	2506	0.997	0.978	16.08	26.96	30.90	71.76
B	Yes	No	1531	2506	0.998	0.995	9.51	11.88	12.72	19.41
C	No	Yes	1785	100	0.995	0.978	15.45	24.16	36.90	70.77
D	Yes	Yes	1531	100	0.998	0.994	9.27	11.89	14.09	21.50

Table 4. Effects of outlier elimination (Layout 3) and dimensionality reduction (GA) on the model performance in predicting **entropy** (J/mol/K).

Configuration	Outlier Elimination	Dimensionality Reduction	Mol.	Desc.	R^2 Train	R^2 Test	*MAE* Train	*MAE* Test	*RMSE* Train	*RMSE* Test
A	No	No	1747	2479	0.983	0.970	14.87	18.71	26.94	35.17
B	Yes	No	1514	2479	0.996	0.995	7.49	8.19	9.90	11.40
C	No	Yes	1747	100	0.980	0.969	14.09	17.37	29.23	35.49
D	Yes	Yes	1514	100	0.996	0.995	7.09	8.01	9.78	11.49

In particular, for the enthalpy, with or without outlier elimination (configurations C and D), the 100 identified descriptors mostly belong to the following categories as defined by AlvaDesc: 2D atom pairs, atom-centered fragments and atom-type e-state indices. For the entropy, the most represented categories in the 100 identified descriptors, that seem to also depend on the outlier elimination step, are the following: 2D atom pairs, 2D autocorrelations and atom-centered fragments (instead of 2D atom pairs, functional group counts and CATS 3D descriptors when outliers are not eliminated). Note that the identified descriptors are fully detailed in the Supplementary Materials, as well as their corresponding coefficients in Lasso linear model. To better understand this variation for the entropy, Table A3 displays the most represented families in the eliminated outliers via Layout 3 for both enthalpy and entropy. For the entropy, the eliminated outliers seem to contain a large percentage of polyfunctional compounds and aromatics, not only as part of the respective individual families, but also as members of some other families, such as esters/ethers and nitrogen compounds (e.g., in esters/ethers, 50% are aromatic and 15% are polyfunctional). Therefore, their elimination could affect their relative importance, as described by the associated descriptors in the dataset. As for CATS 3D descriptors, they consider the spatial pairwise Euclidean distances between potential pharmacophore points or PPP (i.e., hydrogen-bond donor/acceptor, positive/negative, lipophilic) [63]. The elimination of esters/ethers, polyfunctional compounds, nitrogen compounds and aromatics, containing numerous PPP (e.g., the oxygen atom in an ester or ether group is a hydrogen-bond donor), reasonably affects the relevance of CATS 3D descriptors for the remaining data. More generally, these observations demonstrate that the dimensionality reduction step can be influenced by outliers, such as other processing steps. Accordingly, any eventual elimination of outliers should be performed early in the process, as is the case in this work. Other options include the deployment of specific methods that allow feature selection and outlier detection to be performed simultaneously, or robust regression or scaling in the presence of outliers [64–73].

Finally, configuration D seems to be a good compromise between model performance, interpretability and AD size. The parity plots obtained for this configuration are displayed in Figure 13 for the different train/test splits for enthalpy. The respective plots for entropy are shown in Figure A1. For both enthalpy and entropy, most molecules seem to be well predicted, despite some of them deviating from the main diagonal of the parity plots. This is mainly the case for test molecules and those displaying the highest prediction errors are highlighted in red with their identification numbers (ChemID) in Figures 13 and A1.



(**a**) Split 1

(**b**) Split 2

(**c**) Split 3

(**d**) Split 4

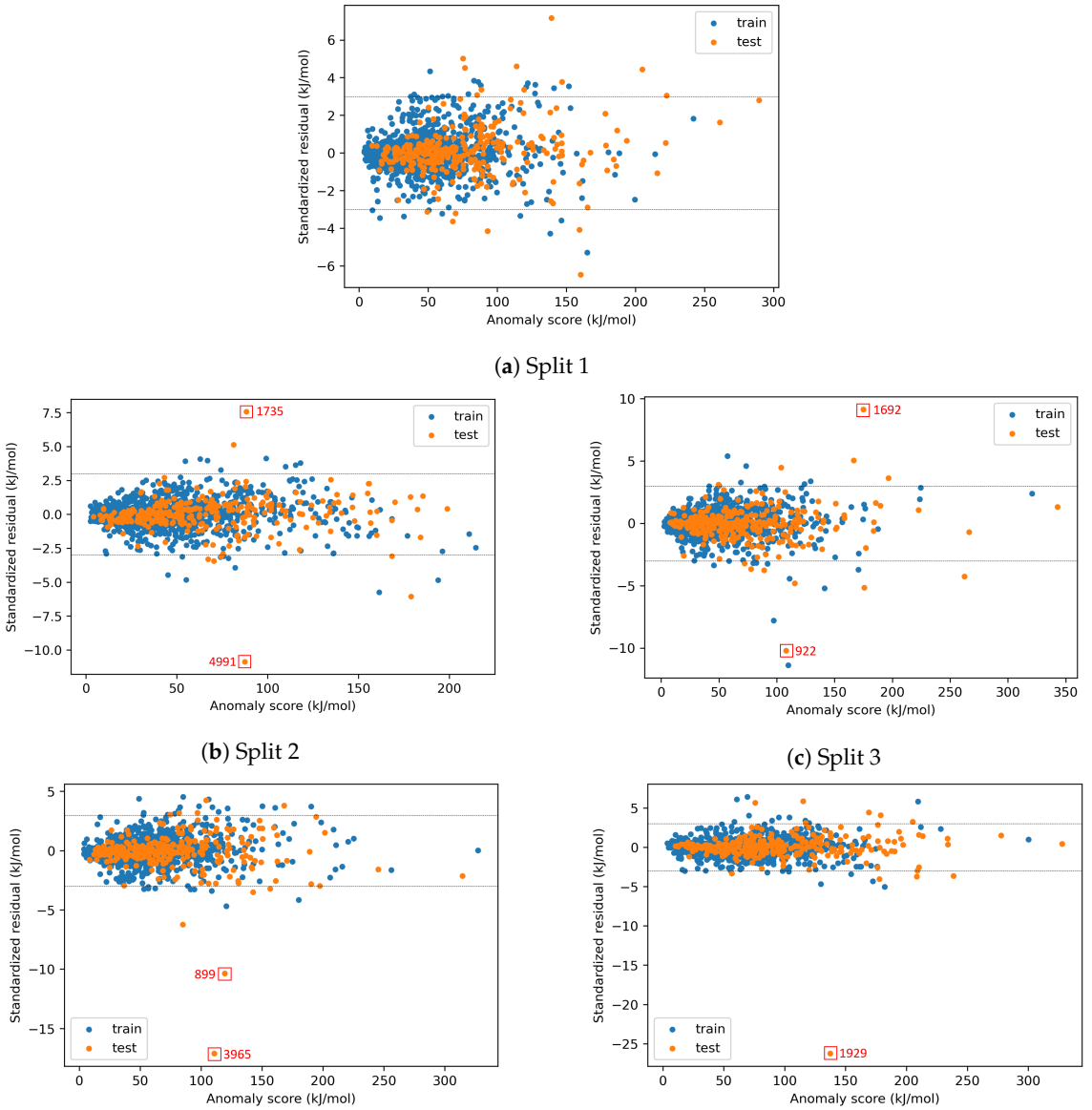
(**e**) Split 5

Figure 13. Parity plots for **enthalpy** after outlier elimination (Layout 3) and dimensionality reduction (GA).

4.2.2. AD Visualization and Evaluation

One of the goals of this second substudy consists of visualizing where the test data are located with respect to the AD defined by the training data. In this sense, AD plots of Lasso standardized residuals versus *RF confidence* anomaly scores are presented in Figures 14 and A2 for the enthalpy and entropy, respectively. These plots correspond to the same splits as the parity plots of Figures 13 and A1. Most test data are located in low anomaly score regions (below 100–150 kJ/mol), similar to the training data. However, a paradox is observed for some points, belonging either to the train or test datasets, which, despite being located in high anomaly score areas, are well predicted by the model. At the same time, other points, located in intermediate anomaly score areas, display poorer predictions. This is, for example, the case for the test molecules with high prediction errors, previously highlighted in the parity plots, which all display intermediate anomaly scores. The origins of these higher prediction errors will be further investigated through tSNE 2D representations in

the next part. More generally, the molecules with high standardized residuals are Model-outliers and their earlier elimination during the preprocessing step could be envisioned as a possible improvement, but under the risk that other molecules become Model-outliers (cf. data-related character of outlier as seen in substudy 1). Additionally, it seems that Model-outliers are more frequent in QSPR/QSAR studies that consider a wide diversity of structures [65], as is the case here, and therefore, building separate models for each category of molecules could also be envisioned, as suggested in the first article [15]. Indeed, the molecules highlighted in red in the parity plots and AD plots correspond to the following chemical families: inorganic, nitrogen, silicon and halogen compounds, and amines/amides. Some of these families were already present during the identification of Model-outliers during the preprocessing stage. This could be an indication that, for some specific families, the model is not able to describe the property value based on the selected descriptors.



(**a**) Split 1

(**b**) Split 2

(**c**) Split 3

(**d**) Split 4

(**e**) Split 5

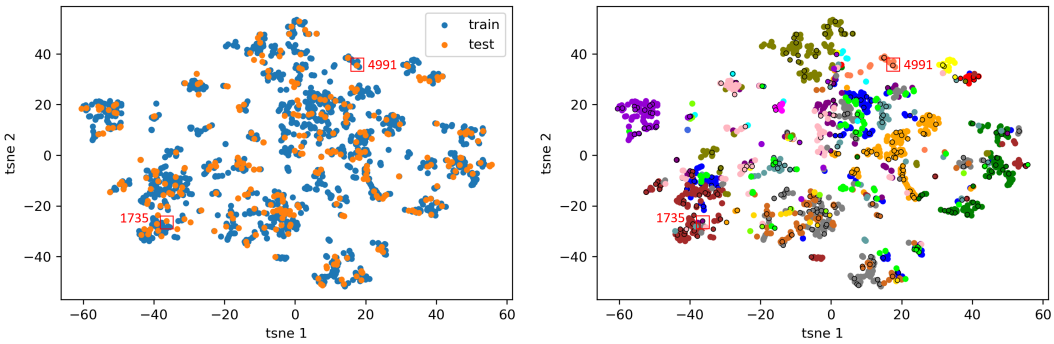
Figure 14. Visualisation of the AD for **enthalpy** after outlier elimination (Layout 3) and dimensionality reduction (GA).

As mentioned earlier, there is no known rule for the limits of the AD, especially regarding the considered method on the x-axis (on the y-axis, the same thresholds as in the leverage method could be considered). Examples of possible thresholds on the x-axis are the maximum anomaly value of training points or a value similar to the upper threshold in the IQR method. In both cases, it is clear that some test points are outside the AD (for x-axis) and should not be used for the model validation (except for testing the extrapolation capacity of the model). This was not further pursued in this work, as most test points were not concerned, but this is a possible improvement. In any case, the elimination of molecules decreases the possibilities of building a generic model.

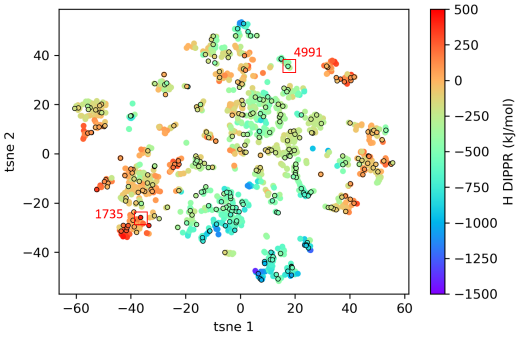
4.2.3. On the Understanding of the High Prediction Errors for Test Molecules

To provide further understanding of the high prediction errors of some test molecules, tSNE 2D is employed to visualize the data in a lower dimensional space. Figures 15a,d and 16a,d correspond to the tSNE 2D representation of splits 2–5 for enthalpy. Figures 15b,e and 16b,e reproduce the previously mentioned subfigures (Figures 15a,d and 16a,d) but while highlighting a different distinctive characteristic of the plotted points. More specifically, these subfigures separate the molecules by chemical family, while those of Figures 15c,f and 16c,f display the reference enthalpy values from the DIPPR database. In all the subfigures, the points with black contours correspond to the test data. The idea behind using all these representations is to find the source of high prediction errors for some test molecules. The hypotheses that are tested here are that these molecules are either located in a low-density area, in terms of the training descriptor space, or they are located in a high-density area but with a high variation in the training property values. Note that tSNE 2D representations of splits 2, 4 and 5 for the entropy are provided in Figures A3 and A4. Also, for Figures 15b,e, 16b,e, A3b,e and A4b, the color codes corresponding to different families are detailed in Figure A5.

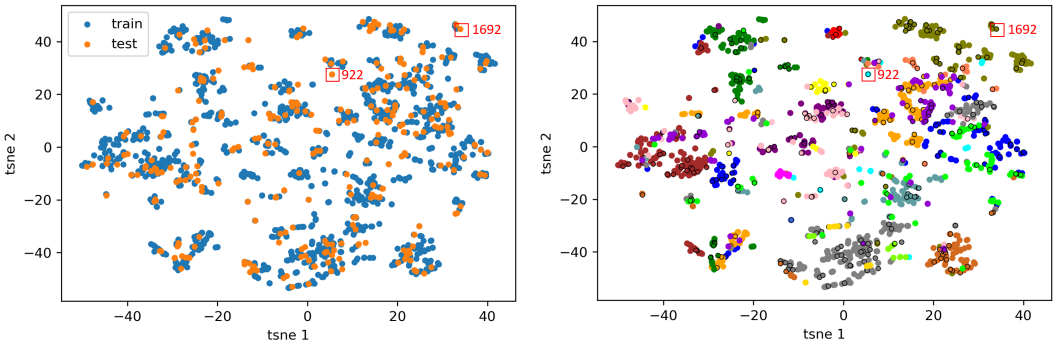
First of all, it is noticeable that the tSNE 2D representation creates some clusters or regions which contain molecules of certain chemical families and/or have property values within a certain range. Indeed, the principle of tSNE is based on the conservation of the local structure of the data, meaning that similar data in a high-dimensional space will be plotted in a close vicinity in a lower-dimensional space. However, the region boundaries are not always clearly delimited and some points do not belong to clusters displaying the above characteristics. Several hypotheses could be put forward to explain this. First of all, the dimensional "compression" of a 100D space to a 2D one could be too high to allow for a clear data clustering. Another reason for the high diversity of chemical families that is observed in some of the formed clusters could be the fact that the selected descriptors (or even the use of molecular descriptors in general) leads to a classification of the molecules that is not completely consistent with their chemical classification in the established chemical families. In other words, molecules may be separated according to their similarities in terms of some descriptor values that would result in new families containing, for example, some alkanes and some esters. Finally, the shape of the clusters in tSNE representation is also highly dependent on the tSNE hyperparameter values, such as the perplexity, which is related to the number of nearest neighbors each point has [56,74].



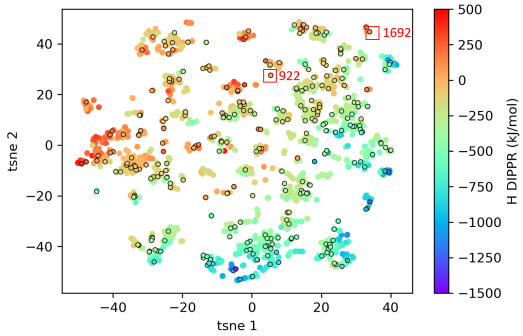
(**a**) Original, split 2 (**b**) Family, split 2



(**c**) Property, split 2

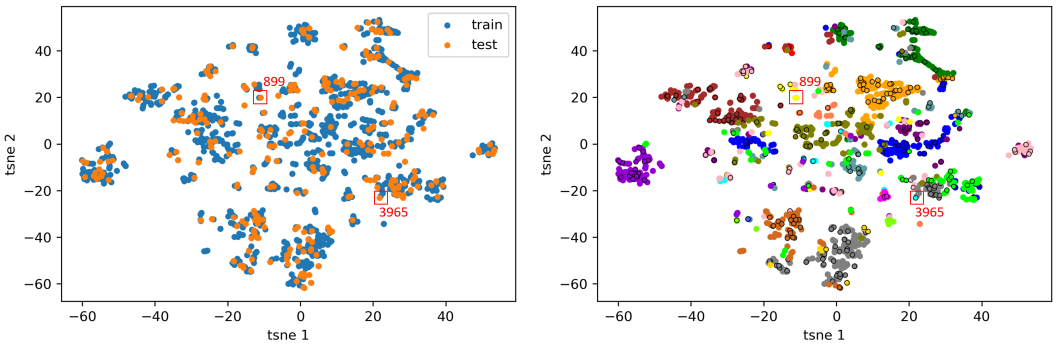


(**d**) Original, split 3 (**e**) Family, split 3



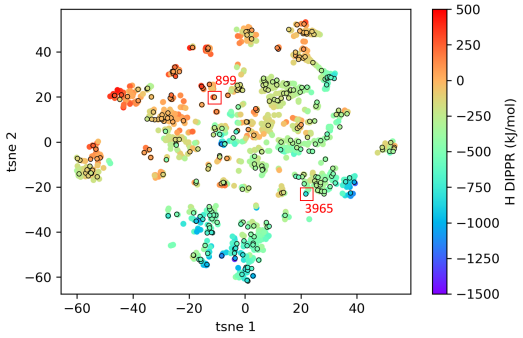
(**f**) Property, split 3

Figure 15. tSNE 2D representations for **enthalpy** after outlier elimination (Layout 3) and dimensionality reduction (GA), **splits 2 and 3**. Subfigures (**a**) and (**d**) show the original data, distinguishing train and test sets. Subfigures (**b**) and (**e**) show the same respective data, distinguishing chemical families (color codes given in Figure A5). Subfigures (**c**) and (**f**) show the same respective data, distinguishing the reference H values from the DIPPR.

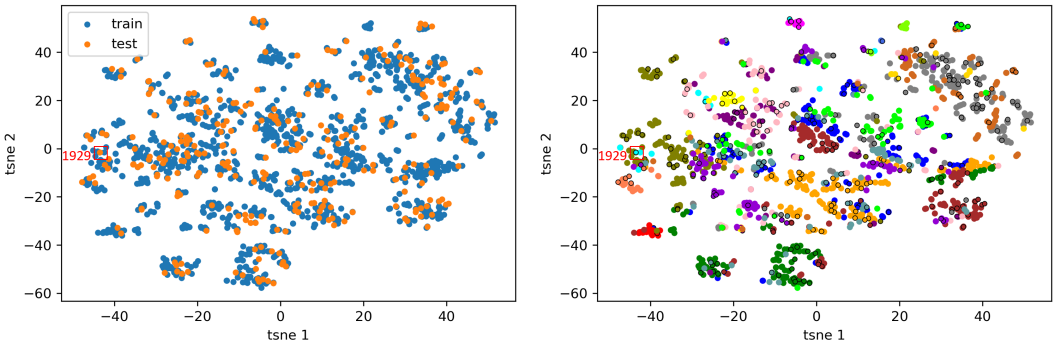


(**a**) Original, split 4

(**b**) Family, split 4

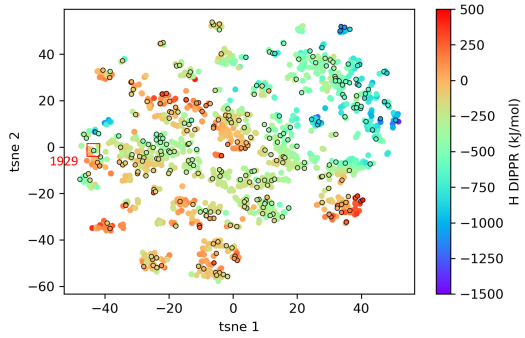


(**c**) Property, split 4



(**d**) Original, split 5

(**e**) Family, split 5



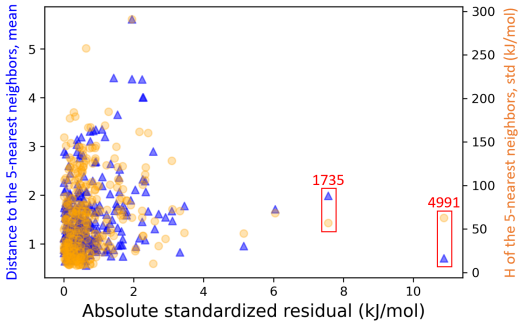
(**f**) Property, split 5

Figure 16. tSNE 2D representations for **enthalpy** after outlier elimination (Layout 3) and dimensionality reduction (GA), **splits 4 and 5**. Subfigures (**a**) and (**d**) show the original data, distinguishing train and test sets. Subfigures (**b**) and (**e**) show the same respective data, distinguishing chemical families (color codes given in Figure A5). Subfigures (**c**) and (**f**) show the same respective data, distinguishing the reference H values from the DIPPR.

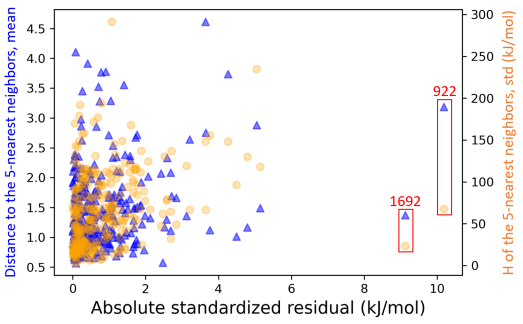
Secondly, the combination of tSNE representations permits us to highlight the potential reasons for the observed high prediction errors of some test molecules. In this sense, one possible explanation—which is, for example, corroborated by the test molecules with the highest prediction errors in splits 4 and 5—is related to the isolation of these molecules from other "similar" ones in the training data. For example, in split 4, the molecules with chemid n°3965 and n°899, which are inorganic compounds, are close only to a few training molecules that are completely different in structure (i.e., esters/ethers and nitriles), thus displaying a significantly different property value. Another similar example concerns the amine molecule n°1735 and the inorganic molecule n°1929 in splits 2 and 5, which are located within a dense region composed of chemically diversified compounds (halogen and inorganic compounds for the first molecule; aromatics, epoxides and esters/ethers for the second one), which have large differences in property values.

Another possible explanation for the observed high prediction errors of some test molecules concerns molecules that are found completely isolated on the graphs, as a result of a low representation of their characteristics (i.e., as perceived by the model on the basis of the used descriptors) in the dataset. An example is the molecule n°922 in Split 3. Finally, it is also observed that some clusters are formed by molecules that display high variation in their enthalpy value, although the value is identified as "similar" by the model. This can lead to underfitting and poor prediction performance in these areas and can be due either to the uncertainties that might exist in the reference property values in the DIPPR database, or to the incapacity of the (selected) descriptors to capture the complete spectrum of structural differences of the molecules that are associated with the measured property. In this sense, different approaches exist in the literature where the use of descriptors is replaced by alternative molecular representation methods, such as graph neural networks (GNN), each one displaying its own advantages and drawbacks [75–77].

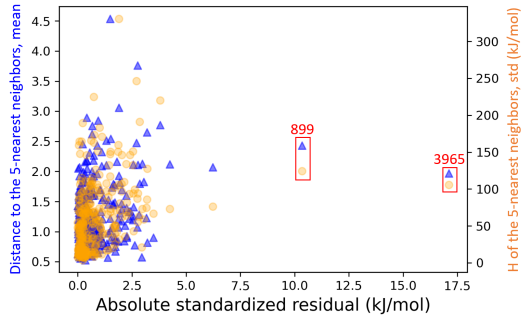
The fact that test molecules, having close training neighbors (in tSNE 2D projection) with low variance in their property values, are generally well predicted can be more clearly visualized in Figure 17. Here, again, the results shown are only for splits 2 to 5 for the enthalpy, but the results for splits 2, 4 and 5 for the entropy are also available in Figure A6. In particular, Figure 17 shows, for each test point, the average distance to its five nearest training neighbors and the standard deviation in the property values of these same neighbors, in relation to the prediction error of the test point. What is observable is that most points are located on the bottom-left corner, which confirms that well-predicted test points are generally those that have close training neighbors with low variance in their property values. However, a high distance to training points and/or a high variance in the property value of closest training points do not always lead to poor predictions, while, in some rare cases, molecules displaying relatively small distance to their five nearest neighbors and small standard deviation in the property values of these neighbors, are poorly predicted by the model. This is, for example, the case of the molecule n°4991 in Split 2 or the molecule n°1692 in Split 3. Further analyses are needed to better explain this contradicting behavior, but the previously emitted hypotheses can also apply here, including the unadapted descriptor-based representation, the imperfect tSNE 2D representation, the lack of training data in some regions, the AD representation ignoring the influence of the descriptors on the model, the presence of property cliffs, or the uncertainties in the DIPPR property values.



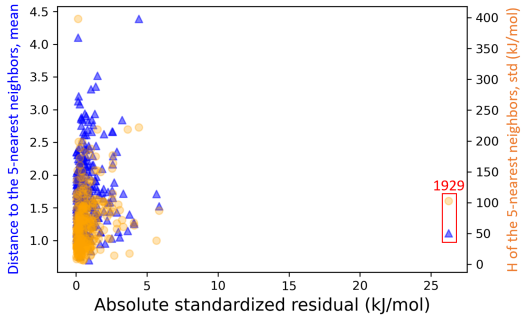
(**a**) Split 2



(**b**) Split 3



(**c**) Split 4



(**d**) Split 5

Figure 17. Analysis of the causes of the high prediction errors for **enthalpy** after outlier elimination (Layout 3) and dimensionality reduction (GA).

4.3. AD Definition during ML Model Deployment (Substudy 3)

In this last substudy, the objective is to show a concrete deployment of the developed ML-QSPR models to new species, including the predictions and the analysis of the reliability of the predictions with respect to the AD. As mentioned earlier, the new species are classified into four categories: A, B, C and D. It is necessary to divide the species into different categories to limit the loss of information. More concretely, one important limitation of descriptor-based approaches is their applicability when at least one descriptor cannot be calculated for the new query species. This is even more likely to happen with the models developed in this work, as they were trained on a large variety of chemical families and they are now being used to screen various types of species. Table 5 shows the number of descriptors that could not be calculated among the 100 selected by the GA method in each category of new species and each of the five train/test splits.

Table 5. Number of non-computable descriptors (among the 100 descriptors selected by GA) for each category of new species.

Property	Category of Species	Split 1	Split 2	Split 3	Split 4	Split 5
H	A	1	0	0	1	2
	B	7	8	5	4	4
	C	18	9	13	12	21
	D	12	16	11	12	9

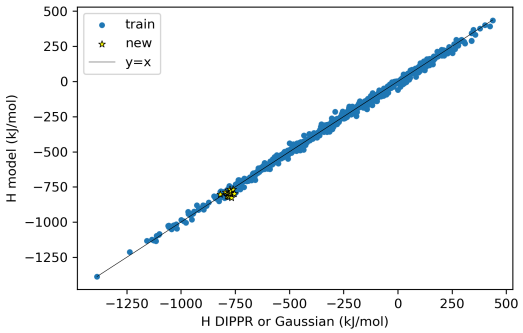
Table 5. *Cont.*

Property	Category of Species	Split 1	Split 2	Split 3	Split 4	Split 5
S	A	9	6	5	9	11
	B	0	0	0	1	1
	C	26	23	26	23	33
	D	25	19	20	20	24

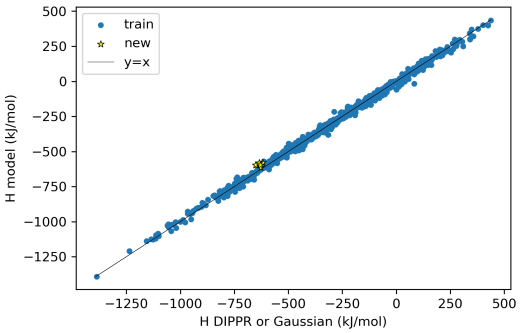
It can be observed that, for some categories of species, the number of impossible-to-calculate descriptors becomes significant before the total number of considered descriptors, reaching percentages in the order of 20% to 25%. This is, for example, the case for categories C and D, especially for the entropy calculation. Indeed, several species in these categories contain a specificity that prevents the calculation of several descriptors. Note that, in the DIPPR data that were used for the model construction, no species with this specificity were present. As such, the 100-GA descriptors, identified via the dimensionality reduction step, do not necessarily represent these species adequately. One possible solution would have been to develop a ML model based on data enriched with similar species as those in categories C and D; however, they were not readily available during the study. Another factor preventing the calculation of certain descriptor values for Cat. C species is also related to the presence of specific atoms [54]. In fact, a posterior analysis has revealed that the few molecules that exist in the DIPPR database, containing these specific atoms, were eliminated during the preprocessing stage, either due to missing property or descriptor values (cf. first article of the series [15]).

Consequently, the ML/QSPR models needed to be retrained on the basis of the proportion of the 100 descriptors, selected using the GA method, which could be calculated for the new species, before implementing the model for the prediction of their properties. This turned out to be problematic in the case of category C, as the significant elimination of descriptors resulted in duplicate molecules in the training set (i.e., molecules that could no longer be differentiated on the basis of the remaining descriptors).

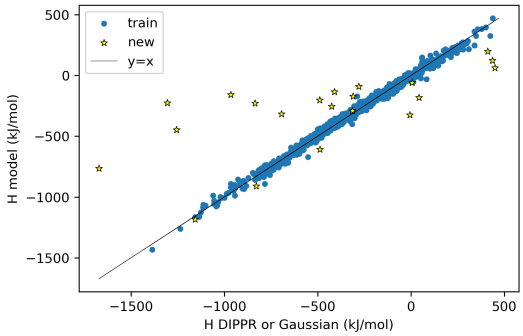
The parity plots for Split 1 are presented in Figures 18 and 19 for the enthalpy and entropy, respectively. In Figure 18c, it is interesting to observe that the points showing higher distances to the y = x line correspond to species containing atoms that were not represented in the training data, as discussed earlier. The detailed predictions of the enthalpy and entropy of all the new species, according to various splits, are provided in the Supplementary Materials. The prediction errors of the new species vary depending on the splits, meaning that some species are better predicted in some splits than in others. In an attempt to assess the reliability of the predictions with respect to the AD, different visualizations are once again attempted in Figures 20–23. In particular, subfigures a and d of these figures display the standardized residuals vs. the *RF confidence* anomaly scores of the training molecules and the new species of Split 1, for the enthalpy and entropy, respectively. Only the new species belonging to Cat. D—and some to Cat. C, in the case of the entropy—seem to have very low *RF confidence* anomaly scores. Therefore, their predicted property values should have a high reliability, and this is effectively the case, as their standardized residuals are also very low. Note that in this work, the standardized residuals for the new species are available (except for some in Cat. C), but in real conditions, this would not have been the case.



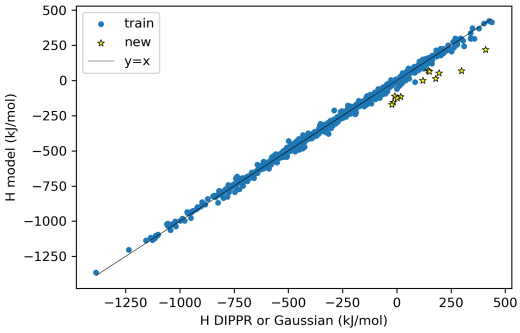
(**a**) Cat. A



(**b**) Cat. B

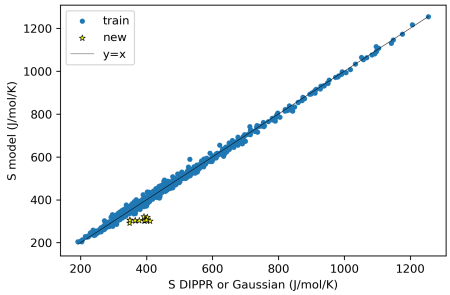


(**c**) Cat. C

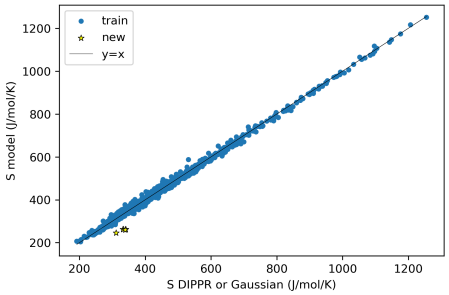


(**d**) Cat. D

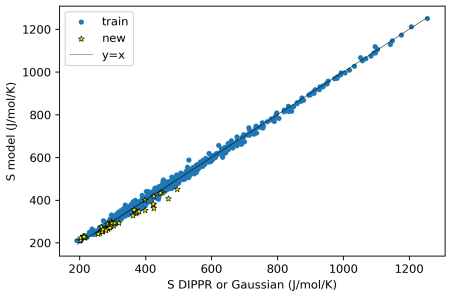
Figure 18. Parity plots of the training molecules and the new species for **enthalpy** for Split 1.



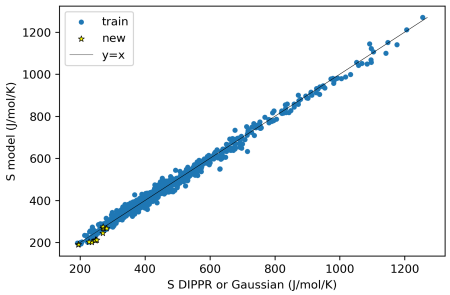
(**a**) Cat. A



(**b**) Cat. B

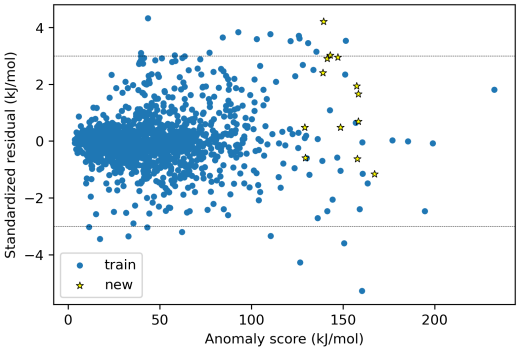


(**c**) Cat. C

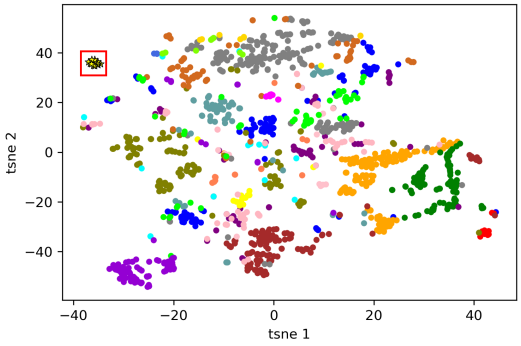


(**d**) Cat. D

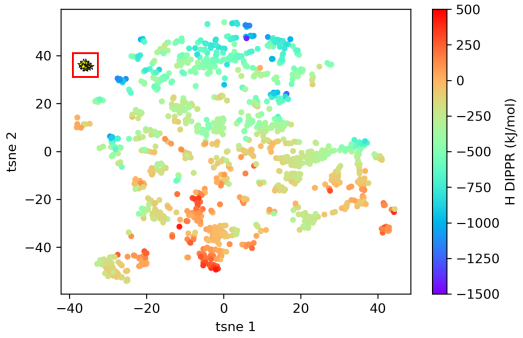
Figure 19. Parity plots of the training molecules and new species for **entropy** for Split 1.



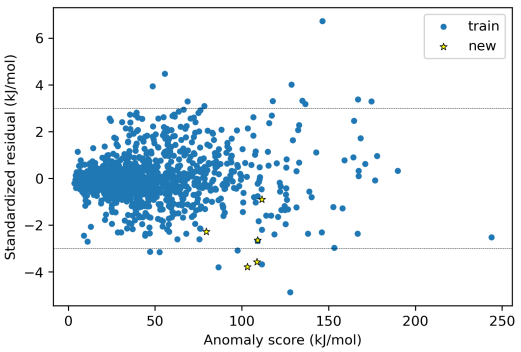
(**a**) Cat. A



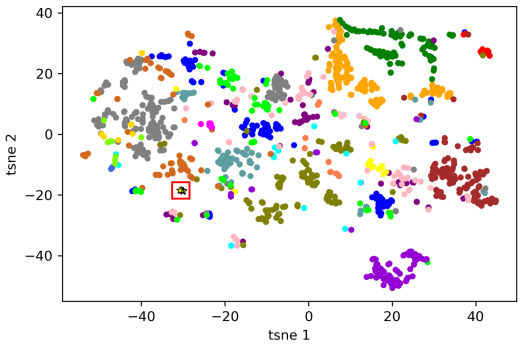
(**b**) Cat. A



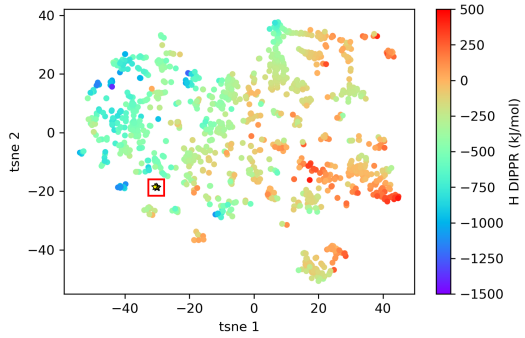
(**c**) Cat. A



(**d**) Cat. B

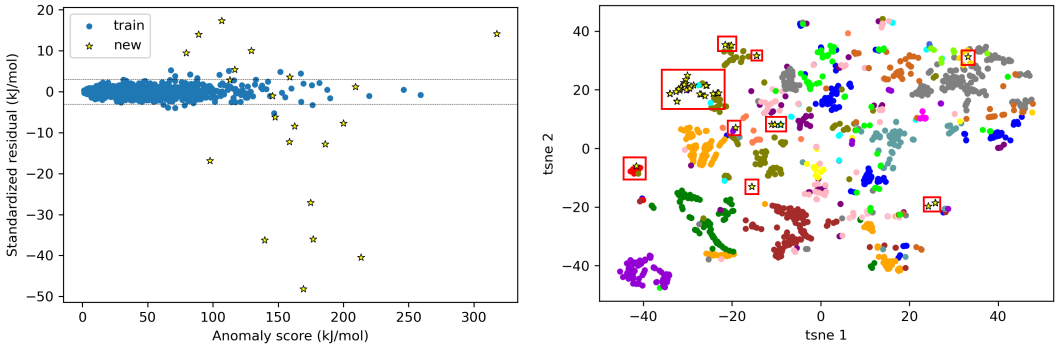


(**e**) Cat. B

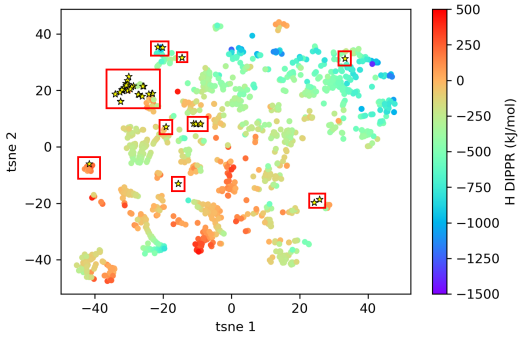


(**f**) Cat. B

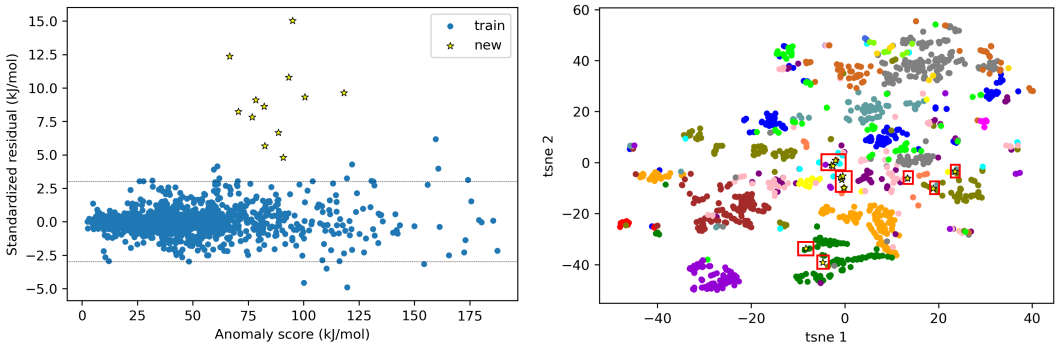
Figure 20. Visualization of the new species in **Cat. A and B** with respect to the AD of the **enthalpy** model for Split 1. Subfigures (**a**) and (**d**) show the standardized residuals vs. the *RF confidence* anomaly scores, distinguishing training data and new species. The other subfigures show the tSNE 2D representation of the same data, distinguishing chemical families (color codes given in Figure A5) for subfigures (**b**) and (**e**), or distinguishing the reference H values for subfigures (**c**) and (**f**).



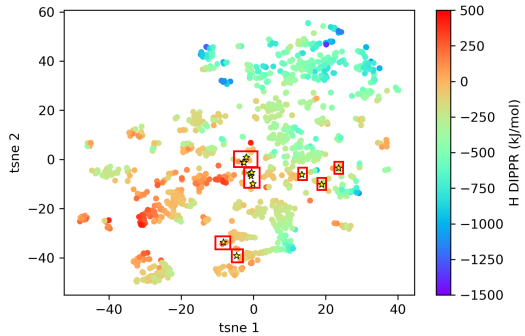
(**a**) Cat. C (**b**) Cat. C



(**c**) Cat. C

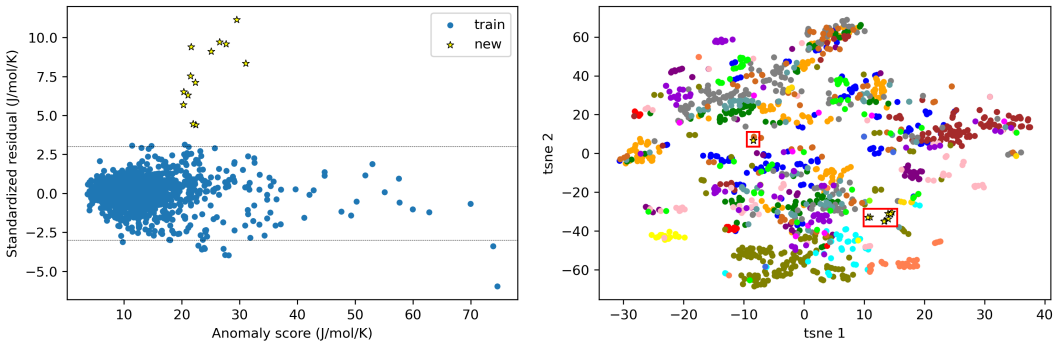


(**d**) Cat. D (**e**) Cat. D

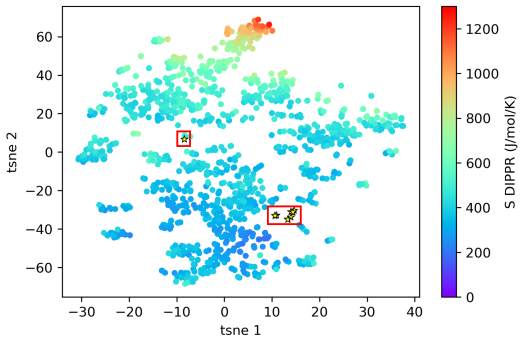


(**f**) Cat. D

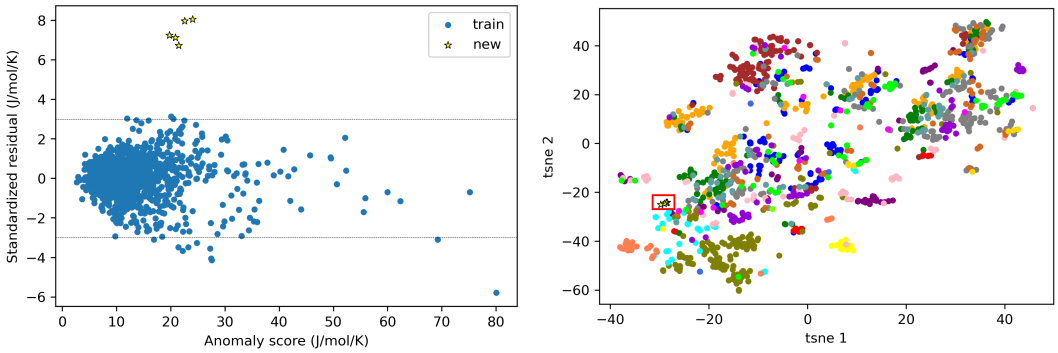
Figure 21. Visualization of the new species in **Cat. C and D** with respect to the AD of the **enthalpy** model for Split 1. Subfigures (**a**) and (**d**) show the standardized residuals vs. the *RF confidence* anomaly scores, distinguishing training data and new species. The other subfigures show the tSNE 2D representation of the same data, distinguishing chemical families (color codes given in Figure A5) for subfigures (**b**) and (**e**), or distinguishing the reference H values for subplots (**c**) and (**f**).



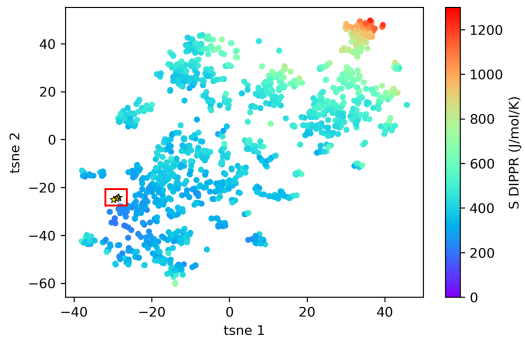
(**a**) Cat. A (**b**) Cat. A



(**c**) Cat. A

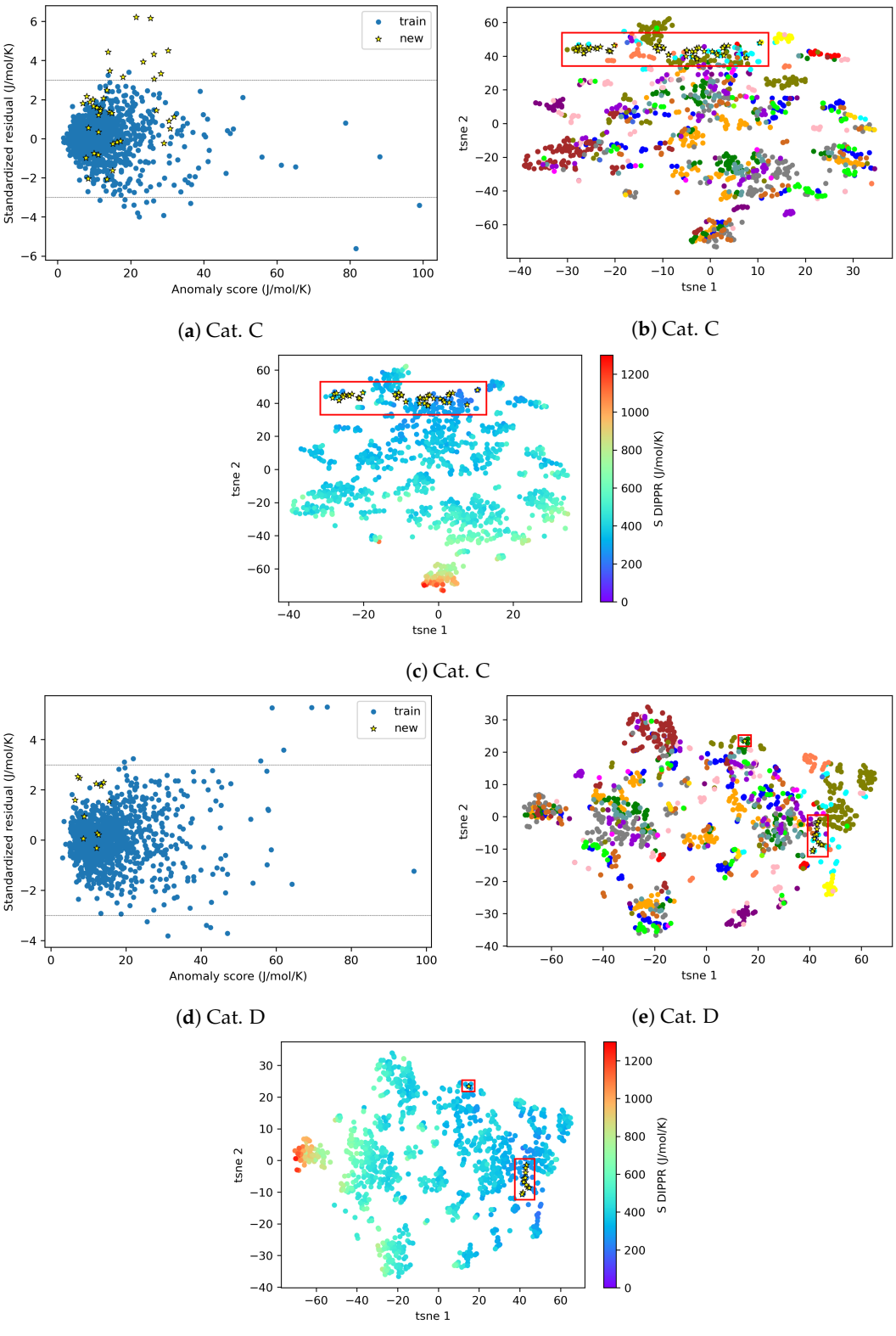


(**d**) Cat. B (**e**) Cat. B



(**f**) Cat. B

Figure 22. Visualization of the new species in **Cat. A and B** with respect to the AD of the **entropy** model for Split 1. Subfigures (**a**) and (**d**) show the standardized residuals vs. the *RF confidence* anomaly scores, distinguishing training data and new species. The other subfigures show the tSNE 2D representation of the same data, distinguishing chemical families (color codes given in Figure A5) for subfigures (**b**) and (**e**), or distinguishing the reference S values for subfigures (**c**) and (**f**).



(**a**) Cat. C

(**b**) Cat. C

(**c**) Cat. C

(**d**) Cat. D

(**e**) Cat. D

(**f**) Cat. D

Figure 23. Visualization of the new species in **Cat. C and D** with respect to the AD of the **entropy** model for Split 1. Subfigures (**a**) and (**d**) show the standardized residuals vs. the *RF confidence* anomaly scores, distinguishing training data and new species. The other subplots show the tSNE 2D representation of the same data, distinguishing chemical families (color codes given in Figure A5) for subplots (**b**) and (**e**), or distinguishing the reference S values for subfigures (**c**) and (**f**).

Conversely, for the rest of the new species and/or for enthalpy, the *RF confidence* anomaly scores are higher and the standardized residuals fluctuate significantly. For example, both very low (e.g., Cat. A or B for enthalpy) and very high (e.g., Cat. C or D for enthalpy, Cat. A or B for entropy) values are observed for the standardized residuals. By highlighting the chemical families and the reference property values (Figures 20b,c,e,f–23b,c,e,f), similarly as in substudy 2, it is possible to observe that the new species that are not well predicted are those located in regions with no—or very few—training points, or those located in higher density regions but with very different property values. However, it remains difficult to understand why some species that have high anomaly scores and/or are isolated from the training data can be either very well predicted (e.g., Cat. A and B in the case of enthalpy) or not (e.g., Cat. A and B in the case of entropy). Further work needs to be conducted to better assess the reliability of the predictions for new species, as the *RF confidence* anomaly score seems not to be a sufficient criterion.

5. Conclusions and Perspectives

In this work, the applicability domain (AD) of machine learning (ML) models trained on high-dimensional data is investigated. The considered models are two ML models developed in a previous article for the prediction of the ideal gas enthalpy of formation and entropy of molecules based on their high-dimensional descriptors [15].

The AD is crucial, as it describes the space of chemical characteristics in which the model can make predictions with a given reliability. This work studies the AD definition of a ML model all along its development procedure: during data preprocessing, model construction and model deployment. Inside each of these steps, the AD definition fulfills a different function. Three AD definition methods, commonly used for outlier detection in high-dimensional problems, were compared: isolation forest (*iForest*), random forest prediction confidence (*RF confidence*) and k-nearest neighbors in the 2D projection of descriptor space obtained via t-distributed stochastic neighbor embedding (*tSNE2D/kNN*). These methods compute an anomaly score which can be used similarly to the distance metrics of classical AD definition methods. A molecule is inside the AD if its distance to the training domain (or anomaly score here) is below a given threshold. Due to the curse of dimensionality, classical AD definition methods are generally unsuitable for high-dimensional problems.

During data preprocessing, the three AD definition methods were used to identify outlier molecules and the effect of their removal was investigated. A more significant improvement in model performance could be observed when outliers identified with *RF confidence* were removed (e.g., for a removal of 30% of outliers, test *MAE* was divided by 2.5, 1.6 and 1.1 for *RF confidence*, *iForest* and *tSNE2D/kNN*, respectively). While these three methods identify X-outliers, the effect of other types of outliers, namely Model-outliers and y-outliers, was also investigated. In particular, the elimination of X-outliers followed by elimination of Model-outliers enabled us to divide MAE and $RMSE$ by two and three, respectively, while reducing overfitting phenomenon. The elimination of y-outliers did not display significant improvement in the model performance.

During model construction, the AD serves to check how validation or test data are located in comparison with training data. All test data were close to training data according to the *RF confidence* method; however, some test data displayed high prediction errors and tSNE 2D representations were used to identify the possible causes (e.g., molecule with chemical information not well represented in training data).

Finally, the developed models were deployed to predict the thermodynamic properties of different categories of molecules. The results show that anomaly scores calculated using the *RF confidence* method are not sufficient to judge the reliability of a prediction for a new molecule. Indeed, some new molecules with low/high anomaly scores were poorly/well predicted. Further research needs to be conducted to identify more appropriate metrics, possibly in combination with the *RF confidence* method, to benefit from the information contained in each metric.

More generally, the AD definition should be better integrated into any QSPR/QSAR procedure for later use of the developed models.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/a16120573/s1, S1 (excel): Data, outliers and ML predictions.

Author Contributions: C.T.: literature review, conceptualization, methodology, data curation and modeling, writing (original draft preparation, review and editing). D.M.: supervision, methodology, writing (review and editing). S.L. and O.H.: data provision, molecular and thermodynamic analyses, generation of data for substudy 3, writing (review and editing). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by MESRI (Ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation), and by the Institute Carnot ICEEL (Grant: "Recyclage de Pneus par Intelligence Artificielle-RePnIA"), France.

Data Availability Statement: The authors do not have the permission to share the data from DIPPR; only some information about the descriptors and the predictions are available in the Supplementary Material (excel). The data of the new species used during model deployment cannot be published.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AD	Applicability domain.
DIPPR	Design institute for physical properties.
DFT	Density functional theory.
H	Enthalpy for ideal gas at 298.15 K and 1 bar.
IQR	Interquartile range.
GA	Genetic algorithm.
GNN	Graph neural network.
GP	Gaussian processes.
iForest	Isolation forest.
kNN	k-nearest neighbors.
Lasso	Least absolute shrinkage and selection operator.
MAE	Mean absolute error.
ML	Machine learning.
OECD	Organisation for economic co-operation and development.
PCs	Principal components.
PCA	Principal component analysis.
QSAR	Quantitative structure–activity relationship.
QSPR	Quantitative structure–property relationship.
R^2	Coefficient of determination.
REACH	Registration, Evaluation, Authorization and Restriction of CHemicals.
RF	Random forest.
RF confidence	Random forest prediction confidence.
$RMSE$	Root mean square error.
S	Absolute entropy of ideal gas at 298.15 K and 1 bar.
tSNE	t-distributed stochastic neighbor embedding.

Appendix A. Identification of Model-outliers and X-outliers in the Preprocessed Data for Enthalpy

Table A1. Top Model-outliers in the preprocessed data for **enthalpy**.

Absolute Standardized Residual r_i	Chemid	Family
$r_i \geq 10$	3608	Halogen Compounds
	2628	Halogen Compounds
	3862	Polyfunctional Compounds
	7886	Polyfunctional Compounds
	7887	Polyfunctional Compounds
$5 \leq r_i < 10$	1840	Sulfur Compounds
	2254	Organic Acids
	1950	Inorganic Compounds
	1969	Silicon Compounds
$3 \leq r_i < 5$	2283	Organic Acids
	3931	Silicon Compounds
	3948	Silicon Compounds
	3929	Silicon Compounds
	4994	Silicon Compounds
	9858	Organic Salts
	2619	Halogen Compounds
	6851	Other Compounds
	2995	Silicon Compounds
	3991	Silicon Compounds
	3958	Silicon Compounds
	2370	Esters/Ethers
	2653	Halogen Compounds
	2877	Polyfunctional Compounds
	3974	Silicon Compounds
	3898	Inorganic Compounds
	6850	Organic Compounds
	9866	Nitrogen Compounds

Table A2. Top 10 X-outliers in the preprocessed data for **enthalpy**.

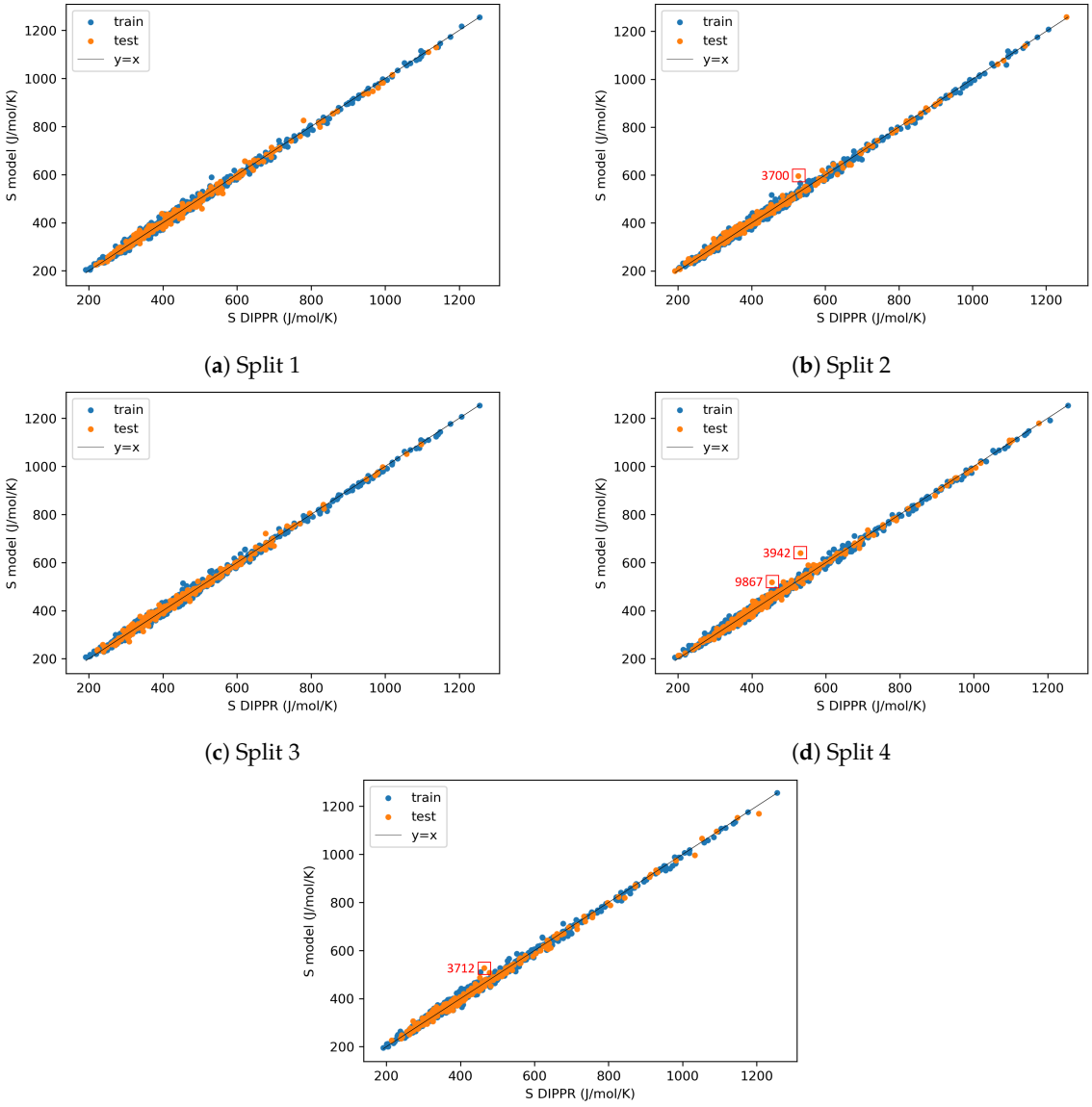
N°	iForest		*RF Confidence*		*tSNE2D/kNN*	
	Chemid	Family	Chemid	Family	Chemid	Family
1	3933	Silicon Compounds	2624	Halogen Compounds	3977	Silicon Compounds
2	3932	Silicon Compounds	3933	Silicon Compounds	1509	Halogen Compounds
3	2624	Halogen Compounds	1627	Halogen Compounds	1866	Halogen Compounds
4	3931	Silicon Compounds	1631	Halogen Compounds	9879	Nitrogen Compounds
5	2991	Silicon Compounds	1626	Halogen Compounds	9877	Nitrogen Compounds
6	1631	Halogen Compounds	3881	Polyfunctional Compounds	90	Alkanes
7	3877	Other Compounds	1930	Inorganic Compounds	5878	Organic Acids
8	2995	Silicon Compounds	1864	Halogen Compounds	1097	Ketones/Aldehydes
9	3348	Esters/Ethers	3932	Silicon Compounds	3056	Ketones/Aldehydes
10	1627	Halogen Compounds	834	Alkanes	7883	Nitrogen Compounds

Appendix B. Most Represented Families in the Eliminated Outliers (Layout 3) for Enthalpy and Entropy

Table A3. Most represented families in the eliminated outliers (Layout 3) for **enthalpy** and **entropy**.

Enthalpy		Entropy	
Halogen Compounds	24%	Esters/Ethers	20%
Silicon Compounds	16%	Silicon Compounds	13%
Esters/Ethers	10%	Polyfunctional Compounds	9%
Nitrogen Compounds	9%	Nitrogen Compounds	9%
Inorganic Compounds	7%	Aromatics	7%
Total	66%	Total	58%

Appendix C. AD Definition for Entropy During Model Construction



(**a**) Split 1

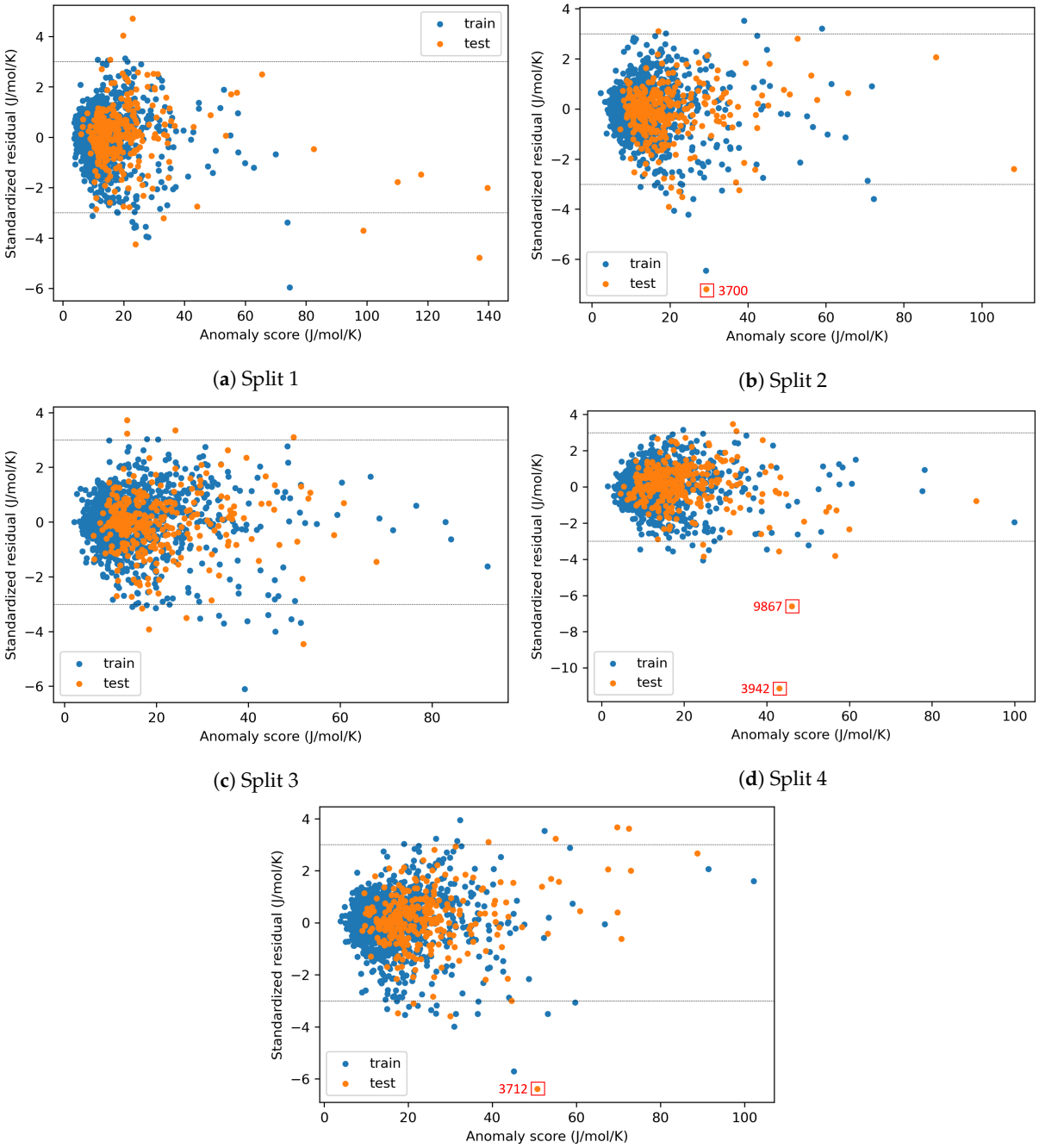
(**b**) Split 2

(**c**) Split 3

(**d**) Split 4

(**e**) Split 5

Figure A1. Parity plots for **entropy** after outlier elimination (Layout 3) and dimensionality reduction (GA).



(**a**) Split 1

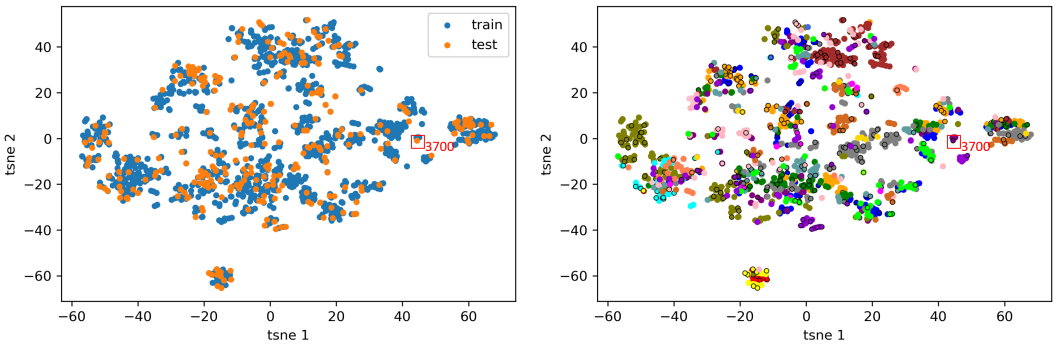
(**b**) Split 2

(**c**) Split 3

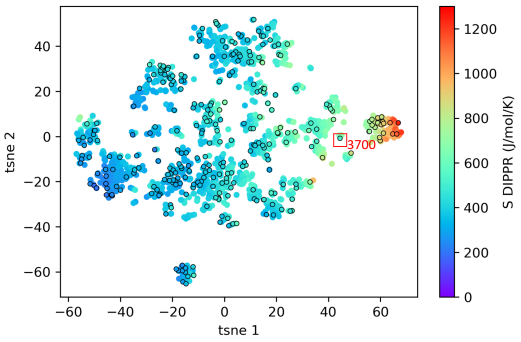
(**d**) Split 4

(**e**) Split 5

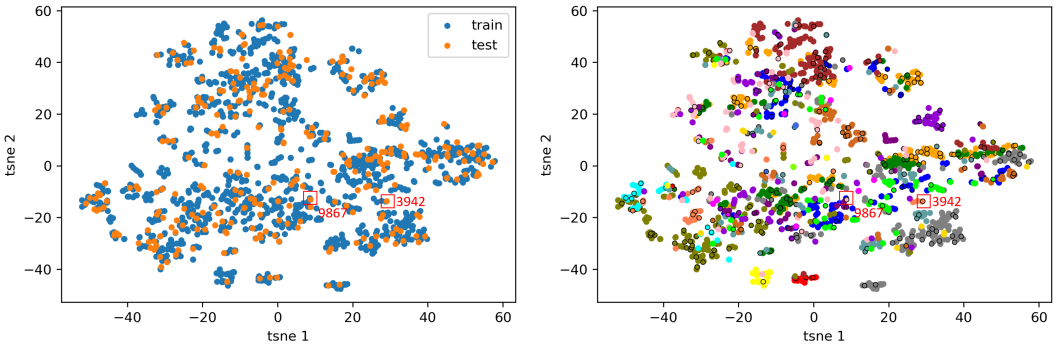
Figure A2. Visualization of the AD for **entropy** after outlier elimination (Layout 3) and dimensionality reduction (GA).



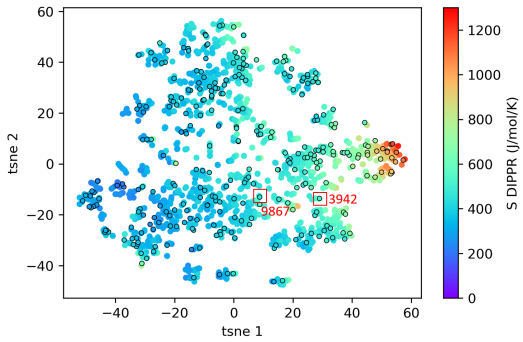
(**a**) Original 2 (**b**) Family 2



(**c**) Property 2

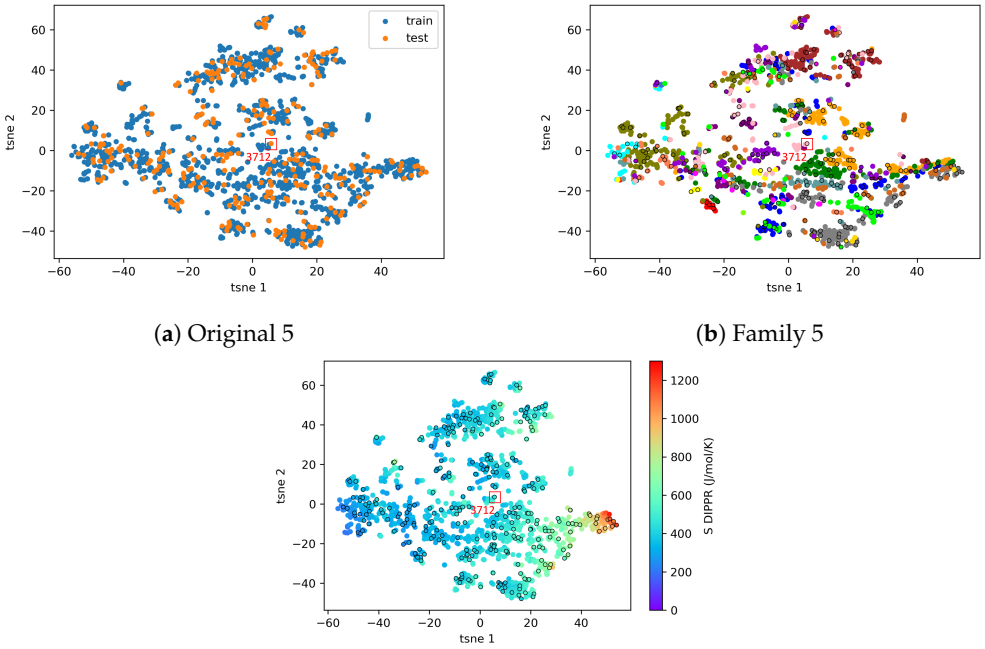


(**d**) Original 4 (**e**) Family 4



(**f**) Property 4

Figure A3. tSNE 2D representations for **entropy** after outlier elimination (Layout 3) and dimensionality reduction (GA), **splits 2 and 4**.



(**a**) Original 5 (**b**) Family 5

(**c**) Property 5

Figure A4. tSNE 2D representations for **entropy** after outlier elimination (Layout 3) and dimensionality reduction (GA), **split 5**.

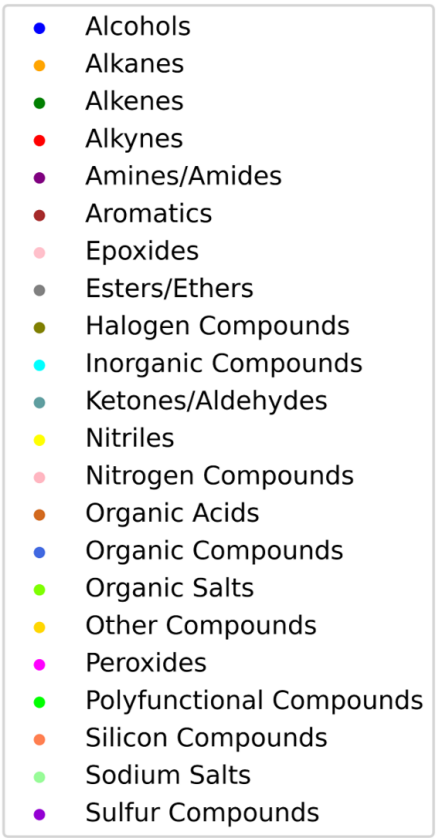
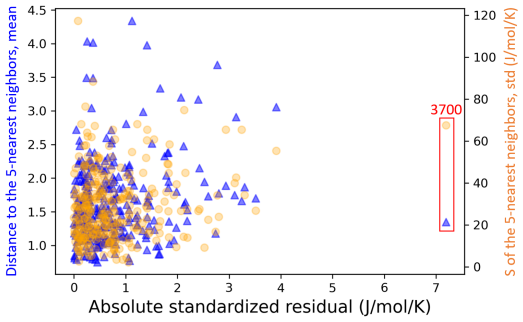
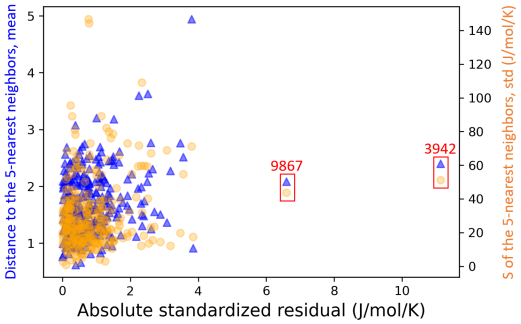


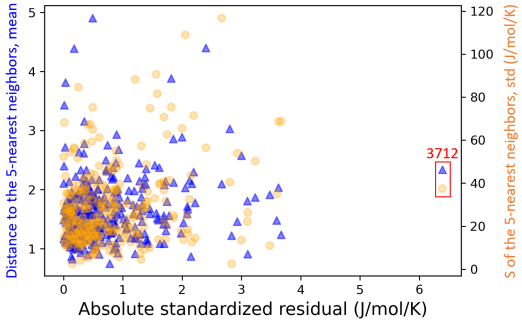
Figure A5. Chemical families considered in Figures 15b,e , 16b,e , A3b,e and A4b .



(**a**) Split 2



(**b**) Split 4



(**c**) Split 5

Figure A6. Analysis of the causes of the high prediction errors for **entropy** after outlier elimination (Layout 3) and dimensionality reduction (GA).

References

1. Netzeva, T.I.; Worth, A.P.; Aldenberg, T.; Benigni, R.; Cronin, M.T.; Gramatica, P.; Jaworska, J.S.; Kahn, S.; Klopman, G.; Marchant, C.A.; et al. Current status of methods for defining the applicability domain of (quantitative) structure-activity relationships. *ATLA Altern. Lab. Anim.* **2005**, *33*, 155–173. [CrossRef]
2. McCartney, M.; Haeringer, M.; Polifke, W. Comparison of Machine Learning Algorithms in the Interpolation and Extrapolation of Flame Describing Functions. *J. Eng. Gas Turbines Power* **2020**, *142*, 061009. [CrossRef]
3. Cao, X.; Yousefzadeh, R. Extrapolation and AI transparency: Why machine learning models should reveal when they make decisions beyond their training. *Big Data Soc.* **2023**, *10*, 20539517231169731. [CrossRef]
4. European Commission Environment Directorate General. *Guidance Document on the Validation of (Quantitative)Structure-Activity Relationships [(Q)Sar] Models*; OECD: Paris, France, 2014; pp. 1–154.
5. Dearden, J.C.; Cronin, M.T.; Kaiser, K.L. How not to develop a quantitative structure-activity or structure-property relationship (QSAR/QSPR). *SAR QSA Environ. Res.* **2009**, *20*, 241–266. [CrossRef] [PubMed]
6. Singh, M.M.; Smith, I.F.C. Extrapolation with machine learning based early-stage energy prediction models. In Proceedings of the 2023 European Conference on Computing in Construction and the 40th International CIB W78 Conference, Crete, Greece, 10–12 July 2023; Volume 4. [CrossRef]
7. Muckley, E.S.; Saal, J.E.; Meredig, B.; Roper, C.S.; Martin, J.H. Interpretable models for extrapolation in scientific machine learning *Digit. Discov.* **2023**, *2*, 1425–1435. [CrossRef]
8. Hoaglin, D.C.; Kempthorne, P.J. Influential Observations, High Leverage Points, and Outliers in Linear Regression: Comment. *Stat. Sci.* **1986**, *1*, 408–412. [CrossRef]
9. Aggarwal, C.C.; Yu, P.S. Outlier detection for high dimensional data. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Santa Barbara, CA, USA, 21–24 May 2001; pp. 37–46. [CrossRef]
10. Akoglu, L.; Tong, H.; Koutra, D. Graph based anomaly detection and description: A survey. *Data Min. Knowl. Discov.* **2015**, *29*, 626–688. [CrossRef]
11. Souiden, I.; Omri, M.N.; Brahmi, Z. A survey of outlier detection in high dimensional data streams. *Comput. Sci. Rev.* **2022**, *44*, 100463. [CrossRef]
12. Smiti, A. A critical overview of outlier detection methods. *Comput. Sci. Rev.* **2020**, *38*, 100306. [CrossRef]

13. Cao, D.S.; Liang, Y.Z.; Xu, Q.S.; Li, H.D.; Chen, X. A New Strategy of Outlier Detection for QSAR/QSPR. *J. Comput. Chem.* **2010**, *31*, 592–602. [CrossRef]
14. De Maesschalck, R.; Estienne, F.; Verdu-Andres, J.; Candolfi, A.; Centner, V.; Despagne, F.; Jouan-Rimbaud, D.; Walczak, B.; Massart, D.L.; De Jong, S.; et al. The development of calibration models for spectroscopic data using principal component regression [Review]. *Internet J. Chem.* **1999**, *2*, 1–21.
15. Trinh, C.; Tbatou, Y.; Lasala, S.; Herbinet, O.; Meimaroglou, D. On the Development of Descriptor-Based Machine Learning Models for Thermodynamic Properties. Part 1—From Data Collection to Model Construction: Understanding of the Methods and their Effects. *Processes* **2023**, *11*, 3325. [CrossRef]
16. Sahigara, F.; Mansouri, K.; Ballabio, D.; Mauri, A.; Consonni, V.; Todeschini, R. Comparison of different approaches to define the applicability domain of QSAR models. *Molecules* **2012**, *17*, 4791–4810. [CrossRef] [PubMed]
17. Jaworska, J.; Nikolova-Jeliazkova, N.; Aldenberg, T. QSAR applicability domain estimation by projection of the training set in descriptor space: A review. *ATLA Altern. Lab. Anim.* **2005**, *33*, 445–459. [CrossRef] [PubMed]
18. Mathea, M.; Klingspohn, W.; Baumann, K. Chemoinformatic Classification Methods and their Applicability Domain. *Mol. Inform.* **2016**, *35*, 160–180. [CrossRef] [PubMed]
19. Roy, K.; Kar, S.; Ambure, P. On a simple approach for determining applicability domain of QSAR models. *Chemom. Intell. Lab. Syst.* **2015**, *145*, 22–29. [CrossRef]
20. Yalamanchi, K.K.; van Oudenhoven, V.C.; Tutino, F.; Monge-Palacios, M.; Alshehri, A.; Gao, X.; Sarathy, S.M. Machine Learning to Predict Standard Enthalpy of Formation of Hydrocarbons. *J. Phys. Chem. A* **2019**, *123*, 8305–8313. [CrossRef]
21. Yalamanchi, K.K.; Monge-Palacios, M.; van Oudenhoven, V.C.; Gao, X.; Sarathy, S.M. Data Science Approach to Estimate Enthalpy of Formation of Cyclic Hydrocarbons. *J. Phys. Chem. A* **2020**, *124*, 6270–6276. [CrossRef]
22. Aldosari, M.N.; Yalamanchi, K.K.; Gao, X.; Sarathy, S.M. Predicting entropy and heat capacity of hydrocarbons using machine learning. *Energy AI* **2021**, *4*, 100054. [CrossRef]
23. Aouichaoui, A.R.; Fan, F.; Abildskov, J.; Sin, G. Application of interpretable group-embedded graph neural networks for pure compound properties. *Comput. Chem. Eng.* **2023**, *176*, 108291. [CrossRef]
24. Balestriero, R.; Pesenti, J.; LeCun, Y. Learning in High Dimension Always Amounts to Extrapolation. *arXiv* **2021**, arXiv:2110.09485.
25. Ghorbani, H. Mahalanobis Distance and Its Application for detecting multivariate outliers. *Ser. Math. Inform.* **2019**, *34*, 583–595. [CrossRef]
26. De Maesschalck, R.; Jouan-Rimbaud, D.; Massart, D.L. The Mahalanobis distance. *Chemom. Intell. Lab. Syst.* **2000**, *50*, 1–18. [CrossRef]
27. Aouichaoui, A.R.; Fan, F.; Mansouri, S.S.; Abildskov, J.; Sin, G. Combining Group-Contribution Concept and Graph Neural Networks Toward Interpretable Molecular Property Models. *J. Chem. Inf. Model.* **2023**, *63*, 725–744. [CrossRef] [PubMed]
28. Mauri, A.; Bertola, M. Alvascience: A New Software Suite for the QSAR Workflow Applied to the Blood–Brain Barrier Permeability. *Int. J. Mol. Sci.* **2022**, *23*, 12882. [CrossRef] [PubMed]
29. Huoyu, R.; Zhiqiang, Z.; Zhanggao, L.; Zhenzhen, X. Quantitative structure–property relationship for the critical temperature of saturated monobasic ketones, aldehydes, and ethers with molecular descriptors. *Int. J. Quantum Chem.* **2022**, *122*, 1–10. [CrossRef]
30. Cao, L.; Zhu, P.; Zhao, Y.; Zhao, J. Using machine learning and quantum chemistry descriptors to predict the toxicity of ionic liquids. *J. Hazard. Mater.* **2018**, *352*, 17–26. [CrossRef]
31. Yousefinejad, S.; Mahboubifar, M.; Eskandari, R. Quantitative structure-activity relationship to predict the anti-malarial activity in a set of new imidazolopiperazines based on artificial neural networks. *Malar. J.* **2019**, *18*, 1–17. [CrossRef]
32. Asadollahi, T.; Dadfarnia, S.; Shabani, A.M.H.; Ghasemi, J.B.; Sarkhosh, M. QSAR models for cxcr2 receptor antagonists based on the genetic algorithm for data preprocessing prior to application of the pls linear regression method and design of the new compounds using in silico virtual screening. *Molecules* **2011**, *16*, 1928–1955. [CrossRef]
33. Kim, M.G. Sources of High Leverage in Linear Regression Model. *J. Appl. Math. Inform.* **2004**, *16*, 509–513.
34. Leys, C.; Klein, O.; Dominicy, Y.; Ley, C. Detecting multivariate outliers: Use a robust variant of the Mahalanobis distance. *J. Exp. Soc. Psychol.* **2018**, *74*, 150–156. [CrossRef]
35. Gramatica, P. Principles of QSAR models validation: Internal and external. *QSAR Comb. Sci.* **2007**, *26*, 694–701. [CrossRef]
36. Varamesh, A.; Hemmati-Sarapardeh, A.; Dabir, B.; Mohammadi, A.H. Development of robust generalized models for estimating the normal boiling points of pure chemical compounds. *J. Mol. Liq.* **2017**, *242*, 59–69. [CrossRef]
37. Sabando, M.V.; Ponzoni, I.; Soto, A.J. Neural-based approaches to overcome feature selection and applicability domain in drug-related property prediction. *Appl. Soft Comput. J.* **2019**, *85*, 105777. [CrossRef]
38. Huang, J.; Fan, X. Reliably assessing prediction reliability for high dimensional QSAR data. *Mol. Divers.* **2013**, *17*, 63–73. [CrossRef] [PubMed]
39. Rakhimbekova, A.; Madzhidov, T.; Nugmanov, R.I.; Baskin, I.; Varnek, A.; Rakhimbekova, A.; Madzhidov, T.; Nugmanov, R.I.; Gimadiev, T.; Baskin, I. Comprehensive Analysis of Applicability Domains of QSPR Models for Chemical Reactions. *Int. J. Mol. Sci.* **2021**, *21*, 5542. [CrossRef] [PubMed]
40. Kaneko, H.; Funatsu, K. Applicability domain based on ensemble learning in classification and regression analyses. *J. Chem. Inf. Model.* **2014**, *54*, 2469–2482. [CrossRef]
41. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
42. Sushko, I. Applicability Domain of QSAR Models. Ph.D. Thesis, Technical University of Munich, Munich, Germany, 2011.

43. Kamalov, F.; Leung, H.H. Outlier Detection in High Dimensional Data. *J. Inf. Knowl. Manag.* **2020**, *19*, 1–15. [CrossRef]
44. Riahi-Madvar, M.; Nasersharif, B.; Azirani, A.A. Subspace outlier detection in high dimensional data using ensemble of PCA-based subspaces. In Proceedings of the 26th International Computer Conference, Computer Society of Iran, CSICC 2021, Tehran, Iran, 3–4 March 2021. [CrossRef]
45. Kriegel, H.P.; Kr, P.; Schubert, E.; Zimek, A. *Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data*; Springer: Berlin Heidelberg, Germany, 2009; Volume 1, pp. 831–838.
46. Filzmoser, P.; Maronna, R.; Werner, M. Outlier identification in high dimensions. *Comput. Stat. Data Anal.* **2008**, *52*, 1694–1711. [CrossRef]
47. Angiulli, F.; Pizzuti, C. Fast outlier detection in high dimensional spaces. In Proceedings of the Principles of Data Mining and Knowledge Discovery, 6th European Conference PKDD, Helsinki, Finland, 19–23 August 2002; pp. 29–41.
48. Kriegel, H.P.; Schubert, M.; Zimek, A. Angle-based outlier detection in high-dimensional data. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 444–452. [CrossRef]
49. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data* **2012**, *6*, 1–39. [CrossRef]
50. Thudumu, S.; Branch, P.; Jin, J.; Singh, J.J. A comprehensive survey of anomaly detection techniques for high dimensional big data. *J. Big Data* **2020**, *7*, 42. [CrossRef]
51. Xu, X.; Liu, H.; Li, L.; Yao, M. A comparison of outlier detection techniques for high-dimensional data. *Int. J. Comput. Intell. Syst.* **2018**, *11*, 652–662. [CrossRef]
52. Zimek, A.; Schubert, E.; Kriegel, H.P. A survey on unsupervised outlier detection in high-dimensional numerical data. *Stat. Anal. Data Min.* **2012**, *5*, 363–387. [CrossRef]
53. Erfani, S.M.; Rajasegarar, S.; Karunasekera, S.; Leckie, C. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognit.* **2016**, *58*, 121–134. [CrossRef]
54. Alvascience, AlvaDesc (Software for Molecular Descriptors Calculation), Version 2.0.8. 2021. Available online: https://www.alvascience.com (accessed on 1 January 2023).
55. Mauri, A. alvaDesc: A tool to calculate and analyze molecular descriptors and fingerprints. *Methods Pharmacol. Toxicol.* **2020**, *2*, 801–820. [CrossRef]
56. van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
57. Gaussian, Inc. *Gaussian 09*; Gaussian, Inc.: Wallingford, CT, USA, 2010.
58. Montgomery, J.A.; Frisch, M.J.; Ochterski, J.W.; Petersson, G.A. A complete basis set model chemistry. VI. Use of density functional geometries and frequencies. *J. Chem. Phys.* **1999**, *110*, 2822–2827. [CrossRef]
59. Becke, A.D. Thermochemistry. III. The role of exact exchange. *J. Chem. Phys.* **1993**, *98*, 5648–5652. [CrossRef]
60. Miyoshi, A. GPOP Software, Rev. 2022.01.20m1. Available online: http://akrmys.com/gpop/ (accessed on 1 January 2023).
61. Non-Positive Definite Covariance Matrices. Available online: https://www.value-at-risk.net/non-positive-definite-covariance-matrices (accessed on 1 June 2023).
62. Cruz-Monteagudo, M.; Medina-Franco, J.L.; Pérez-Castillo, Y.; Nicolotti, O.; Cordeiro, M.N.D.; Borges, F. Activity cliffs in drug discovery: Dr Jekyll or Mr Hyde? *Drug Discov. Today* **2014**, *19*, 1069–1080. [CrossRef]
63. Fechner, U.; Franke, L.; Renner, S.; Schneider, P.; Schneider, G. Comparison of correlation vector methods for ligand-based similarity searching. *J. -Comput.-Aided Mol. Des.* **2003**, *17*, 687–698. [CrossRef]
64. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
65. Cao, D.; Liang, Y.; Xu, Q.; Yun, Y.; Li, H. Toward better QSAR/QSPR modeling: Simultaneous outlier detection and variable selection using distribution of model features. *J.-Comput.-Aided Mol. Des.* **2011**, *25*, 67–80. [CrossRef] [PubMed]
66. Insolia, L.; Kenney, A.; Chiaromonte, F.; Felici, G. Simultaneous feature selection and outlier detection with optimality guarantees. *Biometrics* **2022**, *78*, 1592–1603. [CrossRef] [PubMed]
67. Menjoge, R.S.; Welsch, R.E. A diagnostic method for simultaneous feature selection and outlier identification in linear regression. *Comput. Stat. Data Anal.* **2010**, *54*, 3181–3193. [CrossRef]
68. Kim, S.S.; Park, S.H.; Krzanowski, W.J. Simultaneous variable selection and outlier identification in linear regression using the mean-shift outlier model. *J. Appl. Stat.* **2008**, *35*, 283–291. [CrossRef]
69. Jimenez, F.; Lucena-Sanchez, E.; Sanchez, G.; Sciavicco, G. Multi-Objective Evolutionary Simultaneous Feature Selection and Outlier Detection for Regression. *IEEE Access* **2021**, *9*, 135675–135688. [CrossRef]
70. Park, J.S.; Park, C.G.; Lee, K.E. Simultaneous outlier detection and variable selection via difference-based regression model and stochastic search variable selection. *Commun. Stat. Appl. Methods* **2019**, *26*, 149–161. [CrossRef]
71. Wiegand, P.; Pell, R.; Comas, E. Simultaneous variable selection and outlier detection using a robust genetic algorithm. *Chemom. Intell. Lab. Syst.* **2009**, *98*, 108–114. [CrossRef]
72. Tolvi, J. Genetic algorithms for outlier detection and variable selection in linear regression models. *Soft Comput.* **2004**, *8*, 527–533. [CrossRef]
73. Wen, M.; Deng, B.C.; Cao, D.S.; Yun, Y.H.; Yang, R.H.; Lu, H.M.; Liang, Y.Z. The model adaptive space shrinkage (MASS) approach: A new method for simultaneous variable selection and outlier detection based on model population analysis. *Analyst* **2016**, *141*, 5586–5597. [CrossRef]

74. t-SNE: The Effect of Various Perplexity Values on the Shape. Available online: https://scikit-learn.org/stable/auto_examples/manifold/plot_t_sne_perplexity.html (accessed on 1 June 2023).
75. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4–24. [CrossRef] [PubMed]
76. Xu, K.; Jegelka, S.; Hu, W.; Leskovec, J. How powerful are graph neural networks? In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019; pp. 1–17.
77. Jiang, D.; Wu, Z.; Hsieh, C.Y.; Chen, G.; Liao, B.; Wang, Z.; Shen, C.; Cao, D.; Wu, J.; Hou, T. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *J. Cheminform.* **2021**, *13*, 1–23. [CrossRef] [PubMed]

Chapter 4

Machine Learning modeling of the styrene–GTR radical graft polymerization

Contents

4.1 Introduction

The following two chapters (Chapters 4 and 5) correspond to the second case study investigated during this thesis: the modeling of the styrene-GTR radical graft polymerization.

In 2019, the amount of end-of-life tires was estimated to 30.9 million tons worldwide [86]. Figure 4.1 shows the end-of-life options for discarded tires worldwide in 2019. It can be observed that 59% is properly recycled via various techniques that include material recovery, energy recovery and civil engineering and backfilling. However, 41% is still not correctly recycled and these tires end up for example in landfills, which engenders environmental and health problems. The second case study of this thesis investigates a possible solution to increase the recycling of end-of-life tires. The solution consists, in a first step, in recovering the rubber part of the tires and grinding it into a micrometric powder, also commercially called Ground Tire Rubber or GTR (cf. Figure 4.2). Then, in a second step, the obtained GTR is introduced during the styrene radical polymerization to produce composite materials (polystyrene grafted on GTR, or GTR-graft-PS) with improved mechanical properties in comparison with the brittle pure PS matrix. This polymerization reaction is displayed on Figure 4.3. For this reaction to be optimal, both the styrene conversion rate and the grafting efficiency of PS onto GTR need to be maximized. This case study focuses on the development of ML models (pure ML models in Chapter 4 and hybrid ML/kinetics models in Chapter 5) for predicting the styrene conversion rate as a function of operating conditions. The developed procedure can be applied to the grafting efficiency (or other properties) as a future step of this work.

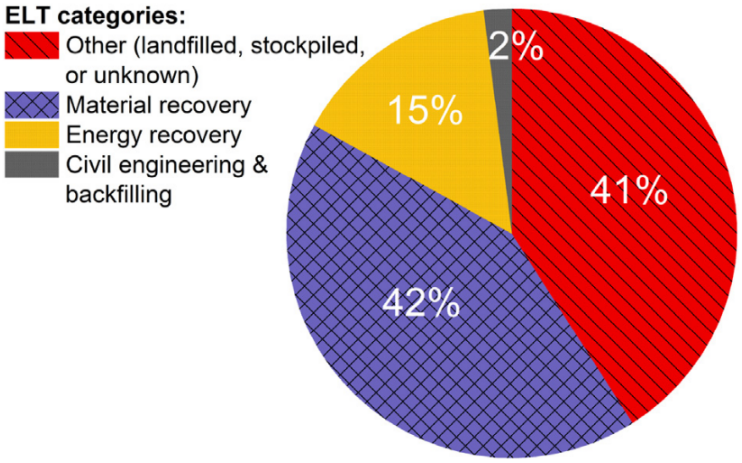


Figure 4.1: End-of-life options for discarded tires worldwide in 2019 [86].

Figure 4.2: Ground tire rubber (GTR).

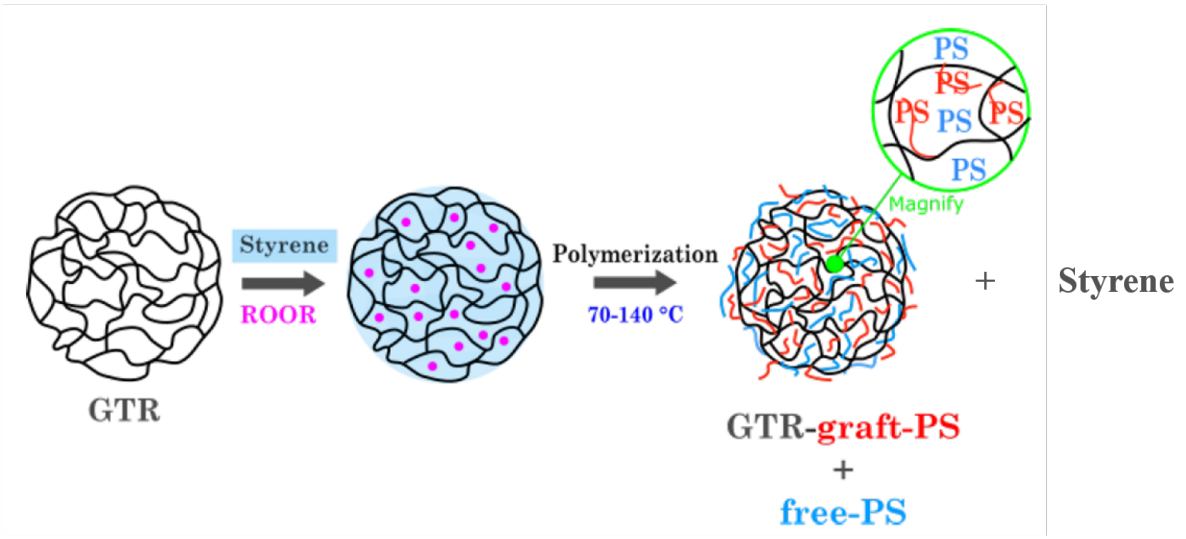


Figure 4.3: Styrene-GTR radical graft polymerization [29].

Why ML?

In this problem, the use of ML is justified by the fact that modeling the kinetics of this system by physico-chemical equations is particularly complex due to the presence of GTR. Indeed, the latter has a complex 3D structure and composition (cf. Figures 4.4 and 4.5). Being derived from different types of tires, the exact composition of the GTR (types/proportions of of elastomers and additives) is not known, which makes it impossible to accurately describe the phenomena and reactions between the different species present in the reactive mixture. GTR and one of its main additives, carbon black, can have a significant impact on the course of the reaction. Figure 4.6 shows the complex 3D structure and the multiple functional groups of carbon black. The latter can more or less interact with the other species (e.g., monomer, initiator) present during the polymerization reaction.

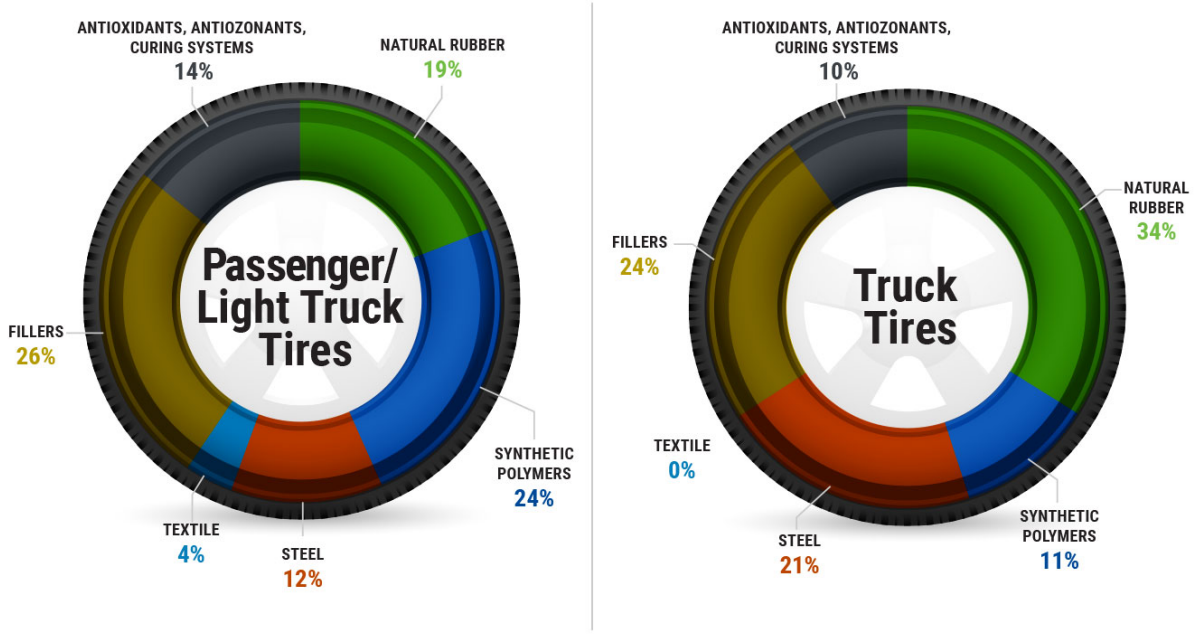


Figure 4.4: Example of composition for two types of tires: passenger/light truck tires and truck tires [3].

Elastomer	Weight fraction
Styrene Butadiene Rubber	40%
Natural rubber	30%
Butadiene rubber	20%
Butyl and nitrile butadiene rubber	10%

Figure 4.5: Example of composition for the elastomer part of the tires [29].

Specificity of the work

The data set characteristics in this case study are completely different from those of the first case study (cf. Chapters 2 and 3), as the available data is limited (as generated by time-consuming experiments) and low dimensional (prevailing operating conditions). But above all, the phenomena occurring in the system can here be partly described by knowledge-based models from the literature. In Chapter 4, an experimental procedure is developed to generate the data necessary to ML modeling and different pure ML approaches are subsequently trained. In Chapter 5, a hybrid scheme, combining Gaussian Processes ML models with knowledge-based models, is implemented to exploit the information in both the process data and knowledge. In particular, the obtained hybrid model benefits from the advantages of each of its components.

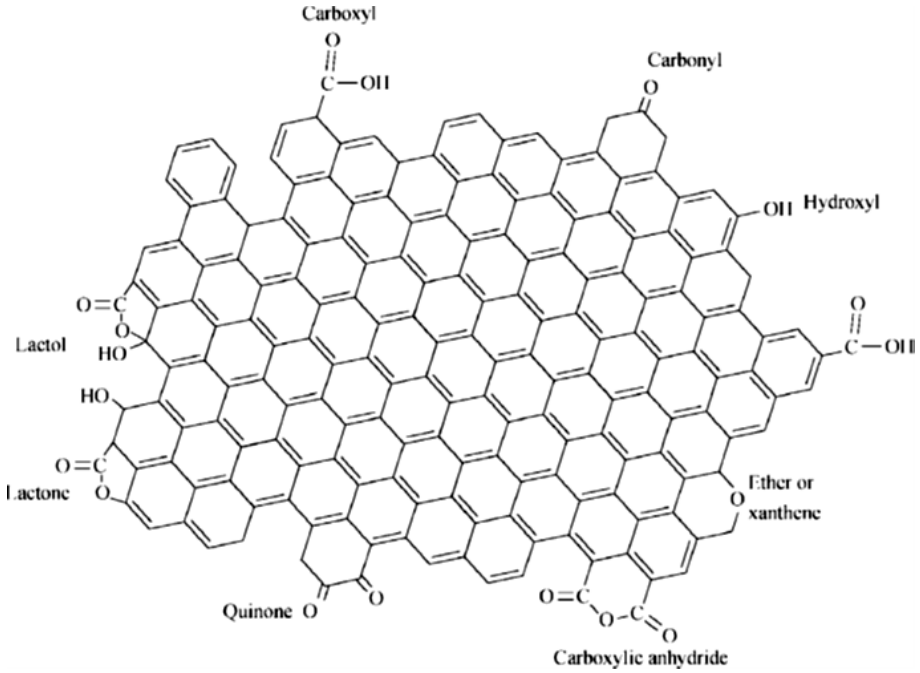


Figure 4.6: Carbon black 3D structure and functional groups [58].

4.2 Outline of the publication

1. **Introduction**
2. **Experimental Section**
   - 2.1 Materials
   - 2.2 Small-Sized System
   - 2.3 Medium-Sized System
   - 2.4 Design of Experiments
3. **ML Modeling**
   - 3.1 ML Algorithms
   - 3.2 ML Procedure
4. **Results and Discussion**
   - 4.1 Experimental Results
   - 4.2 ML Results
5. **Conclusions**

Abbreviations

Appendix A. Additional Photos of the Experimental Medium-Sized System

Appendix B. Evaluation of the Experimental Uncertainties

Appendix C. Detailed Experimental Data for the Small- and Medium-Sized Systems

Appendix D. Detailed ML Results

Appendix E. Ways of Improvement for the ML Model

References

4.3 Publication *"A Comprehensive Study on the Styrene–GTR Radical Graft Polymerization: Combination of an Experimental Approach, on Different Scales, with Machine Learning Modeling", published on 22 February 2023*

macromol

MDPI

Article

A Comprehensive Study on the Styrene–GTR Radical Graft Polymerization: Combination of an Experimental Approach, on Different Scales, with Machine Learning Modeling

Cindy Trinh, Sandrine Hoppe, Richard Lainé and Dimitrios Meimaroglou *

Laboratoire Réactions et Génie des Procédés, Université de Lorraine, CNRS UMR7274, LRGP, F-54000 Nancy, France

* Correspondence: dimitrios.meimaroglou@univ-lorraine.fr

Abstract: The study of the styrene–Ground Tire Rubber (GTR) graft radical polymerization is particularly challenging due to the complexity of the underlying kinetic mechanisms and nature of GTR. In this work, an experimental study on two scales (∼10 mL and ∼100 mL) and a machine learning (ML) modeling approach are combined to establish a quantitative relationship between operating conditions and styrene conversion. The two-scale experimental approach enables to verify the impact of upscaling on thermal and mixing effects that are particularly important in this heterogeneous system, as also evidenced in previous works. The adopted experimental setups are designed in view of multiple data production, while paying specific attention in data reliability by eliminating the uncertainty related to sampling for analyses. At the same time, all the potential sources of uncertainty, such as the mass loss along the different steps of the process and the precision of the experimental equipment, are also carefully identified and monitored. The experimental results on both scales validate previously observed effects of GTR, benzoyl peroxide initiator and temperature on styrene conversion but, at the same time, reveal the need of an efficient design of the experimental procedure in terms of mixing and of monitoring uncertainties. Subsequently, the most reliable experimental data (i.e., 69 data from the 10 mL system) are used for the screening of a series of diverse supervised-learning regression ML models and the optimization of the hyperparameters of the best-performing ones. These are gradient boosting, multilayer perceptrons and random forest with, respectively, a test R^2 of 0.91 ± 0.04, 0.90 ± 0.04 and 0.89 ± 0.05. Finally, the effect of additional parameters, such as the scaling method, the number of folds and the random partitioning of data in the train/test splits, as well as the integration of the experimental uncertainties in the learning procedure, are exploited as means to improve the performance of the developed models.

Keywords: radical graft polymerization; styrene; ground tire rubber; artificial intelligence; machine learning

Citation: Trinh, C.; Hoppe, S.; Lainé, R.; Meimaroglou, D. A Comprehensive Study on the Styrene–GTR Radical Graft Polymerization: Combination of an Experimental Approach, on Different Scales, with Machine Learning Modeling. *Macromol* **2023**, *3*, 79–107. https://doi.org/10.3390/macromol3010007

Academic Editor: Ana María Díez-Pascual

Received: 19 December 2022
Revised: 14 February 2023
Accepted: 17 February 2023
Published: 22 February 2023

1. Introduction

As the consequences of climate change have been more and more visible in recent years, it has become of prior importance for all sectors to reduce the environmental impact of their activities, which is also encouraged by the increasing number of policies going in this direction. In this sense, greener ways to deal with the increasing amount of end-of-life tires are needed to avoid their ending in landfills or burning for energy recovery [1–4].

One environmentally-friendly and popular solution consists of transforming the rubber parts of tires into a micrometric powder, commercially called Ground Tire Rubber (GTR), by mechanical grinding. This powder is subsequently used as raw material or additive in other products of interest for diverse applications. In particular, the elastomeric properties of GTR make it an interesting filler agent to improve the mechanical properties of a wide range of materials including thermoplastics, thermosets, virgin or composite

rubbers, concrete, bitumen or asphalt [1–3,5–9]. For example, the introduction of GTR fillers in polystyrene (PS), which is investigated in this work, can produce a composite material with improved stress cracking resistance and impact strength, with respect to the brittle pure PS matrix, and with reduced material costs.

However, the different structures of the two phases (i.e., the PS matrix and GTR, GTR having the form of a 3D crosslinked network) present a challenge in terms of their compatibility and the resulting poor mechanical properties of the final blend. Accordingly, several compatibilization strategies can be envisioned, such as devulcanization and reclamation or surface activation [2,3,5,6,10,11]. In this work, a chemical surface activation strategy in the form of in situ cross-linking by grafting polymerization, which is among the most prominent GTR surface modification techniques [6], is employed to produce PS-GTR composites. More specifically, PS is directly grafted onto the surface of GTR particles that are present during the radical polymerization of styrene. Similar in situ bulk radical polymerization techniques have been used to finely disperse and incorporate not only rubber particles in polymer matrices [4,12–14], but also other nanofillers, such as for the synthesis of graphene-based nanocomposite materials [15–21], carbon tube-based nanocomposite materials [22,23] and many others [24–26].

A second challenging aspect of this process lies in the complexity of the GTR itself as raw material [2]. GTR is effectively obtained from tires of diverse origins and/or with unknown compositions of rubbers and additives, depending on the vehicle type and the required performance. This limits the mastery of the process and reduces the capacity of a detailed modeling of the system to better understand the link between the process parameters (e.g., temperature/time of the reaction and initial compositions of reactants) and the final productivity and product quality indicators (e.g., styrene conversion rate and grafting efficiency). For example, a previous work demonstrated that additives and moieties found in varying compositions in GTR, such as carbon black, display a significant effect on the course of the polymerization [27]. Nevertheless, developing a model capable of handling this inherent system complexity and predicting the effects of the process conditions on the desired indicators of the produced GTR-based composite would be extremely beneficial to the associated recycling and circular economy sectors.

This work builds on previously reported results of an ongoing study on the styrene–GTR radical graft polymerization system [27–29]. More specifically, a part of this ongoing study has already focused on the modeling of the system via the introduction of a generalized kinetic scheme that has been employed to describe the evolution of the polymerization in the presence of different amounts of GTR and/or benzoyl peroxide (BPO) initiator, under different reaction temperatures [29]. The mathematical model proposed in that work, presented an initial framework for the consideration of the chemical reactions that might occur in the system, as a consequence of the presence of GTR (i.e., as opposed to the presence of pure polybutadiene rubber that is considered in most reported studies [30–33]) as well as a methodology for the quantification and the gradual "masking" of reaction sites on the surface of GTR. In the present work, a complementary approach is adopted as means to circumvent the manque of knowledge on the exact composition of the system and the associated chemical developments. Accordingly, the present study focuses exclusively on the case where GTR is present in the system, contrary to [29] where both the pure styrene homopolymerization and the GTR-grafting cases were investigated.

Machine learning (ML) is a prominent tool in polymer materials design and property prediction that allows saving significant time and effort, related to the development of purely phenomenological modeling approaches, when the underlying phenomena are not completely elucidated and/or highly complex [34,35]. Accordingly, this work presents the implementation of a data-driven modeling approach, on the basis of supervised ML techniques, under varying operating conditions. The developed ML models are implemented as standalone regression models, for the prediction of the monomer conversion developments in an attempt to assess their suitability in modeling this highly complex polymerization system, without using any prior knowledge or mechanistic description of

the underlying phenomena, as it is the case in previous works [27–29]. As a perspective for future work, the findings of this work will be used in the development of a hybrid modeling approach where both phenomenological and ML models of the system will be combined.

The proposed ML modeling framework necessitates the existence of experimental data to serve during the training of the models and the subsequent testing of their generalization capacity. However, the use of the previously reported bench-scale experimental setup [28] was prohibitive for the production of sufficient amounts of data for the ML model (i.e., under a wide range of process conditions), due to its highly time-consuming nature. In addition, this system also presented a limited capacity of controlling the homogeneity of temperature and composition within the reactor, especially at high GTR loading [28]. So in order to produce the required experimental data for this work, with the lowest possible uncertainty and within a reasonable amount of time, the grafting polymerization experiments were performed on laboratory-scale experimental setups, namely a small-sized system ($\sim$10 mL) and a medium-sized one ($\sim$100 mL). In the small-sized system, a parallel realization of grafting polymerizations in sealed test tubes, placed in a heating bath and without internal agitation, was carried out. On the other hand, the medium-sized system was based on an ad hoc experimental setup allowing the simultaneous realization of the poymerization reaction in six glass reactors, under controlled temperature and internal agitation. Both systems served to produce multiple samples rapidly and to investigate the effect of scale-up on the thermal and compositional homogeneity within the reacting mixture. In both cases, small volumes of reactants were used in order to allow using the complete reactor content in the post-reaction analyses, thus limiting the uncertainties related to the sampling of the heterogeneous mixture.

Note that, most reported studies are either limited to low GTR loadings (i.e., up to 30% wt.) or concern different polymer matrices and/or compatibilization strategies [10,12,36–42]. At the same time, the precision and repeatability of the results are rarely discussed. To the authors' knowledge, this is the first time that the styrene–GTR radical graft polymerization system has been comprehensively investigated, for a wide range of GTR content, by combining a two-scale experimental approach with a ML modeling strategy. All the produced experimental data, as well as the developed mathematical models, are readily available through a GitHub repository. In addition, an extensive analysis of the uncertainty introduced in the different steps of the experimental process is also included in the discussion of the results.

2. Experimental Section

2.1. Materials

The reagents used for the polymerization reactions are styrene monomer Reagent-Plus® (with a purity $\geq$ 99% and stabilized with 4-tert-butylcatechol) and BPO initiator Luperox® A75. They were purchased from Sigma-Aldrich Chemie GmbH (Steinheim, Germany) and Sigma-Aldrich Co. (St. Louis, MO, USA) respectively and both were used without further purification. GTR particles were obtained from DeltaGom France and used without further purification as well. The characteristics of GTR have been discussed elsewhere [27,28]. Even if the presence of stabilizers and impurities can greatly affect polymerization kinetics, these were not removed within the perspective of implementing this process at a larger scale. Therefore, the idea in this work was to use GTR as-received without further purification with organic solvents to avoid the generation of chemical wastes and to reduce production costs.

2.2. Small-Sized System

On the small scale, reactions took place in glass test tubes containing a total of 1.5 g of reactants (i.e., styrene, BPO and GTR) at various proportions. The test tubes were first filled with the desired amounts of GTR. The mixtures of monomer and initiator were preliminary prepared in beakers and then introduced directly onto the GTR, in the corresponding test tubes, followed by sealing of the tubes. Once the oil bath reached the target temperature,

about ten test tubes were introduced in the bath and kept submerged during the whole reaction time. Occasionally, the sealing of the tubes was loosened to allow pressure relief in case of gas buildup. At the end of each reaction, the test tubes were immediately placed into a cold water bath for cooling. Due to the very low volumes, the heating and cooling times of the tubes was limited to a few minutes (i.e., typically less than three minutes). Finally, the test tubes were unsealed and placed in vacuum oven at 40 °C until constant weight, signifying the complete evaporation of any remaining styrene. The measured mass loss during this stage served in the calculation of styrene conversion, according to:

$$X_{styrene} = \frac{m_{i,styrene} - m_{f,styrene}}{m_{i,styrene}} \quad (1)$$

where $m_{i,styrene}$ denotes the mass of styrene that was introduced in the test tube and $m_{f,styrene}$ is the recorded mass loss during vacuum evaporation.

2.3. Medium-Sized System

Figure 1 depicts a schematic of the experimental system, which was designed to perform simultaneously six polymerization reactions under identical heating, stirring, inerting and refrigeration conditions. More precisely, Figure 1a corresponds to a side view of the system (i.e., showing only three of the six reactors), while the top view proposed in Figure 1b enables to better visualize all six positions. Besides these schematics, photos of the actual system are provided in Appendix A. In the following, the elements of the system are described.

- The reactors were designed with a limited volume (<100 mL) to keep the duration of sample analysis after polymerization reasonable (note that the whole reactor content is analyzed) while enabling reactions with sufficient amounts of GTR. The collars are large enough to facilitate the introduction of the reactants and the recovery of the products, the latter being more or less viscous depending on the operating conditions. The reactor diameter remains constant along the reactor height (except the bottom). This facilitates the introduction and removal of the agitator. The reactor material (glass) facilitates the observation of the mixing conditions. Finally, a lateral orifice enables the punctual introduction of a thermocouple, inside the reactor, for temperature measurement.
- A 6-cm thick aluminum plate, containing 6 drilled reactor holdings, was used for reactor temperature control (Figure 1b,c). This plate was heated via conduction by an electrical heating plate on which it was directly placed. Silicon oil was also added in the holding positions to maximize the heat-transfer between the aluminum plate and the reactors.
- A cooling system was also installed at the upper part of the setup, enabling to cool down the vapors of monomer during the reaction. It was composed of a glass tube, positioned on top of each reactor and wrapped with a transparent hose in which circulated glycerol, at a temperature of to 3.5 °C, from a cooling thermostat bath.
- The mechanical agitator (Figure A1c) was designed with double propellers and an anchor to better scrape the reacting mixture from the walls and bottom of the reactors. In fact, since the mixture of GTR with PS became quite sticky during the polymerization, it was important to make sure that reactor content would remain under mixing throughout the polymerization, without forming an inverse bell shape with the agitator spinning in void in the middle of it. The rotation speed was fixed at 30 rpm. A pulley was fixed on the top of each stirring axe and a belt system made the 6 stirring axes rotate simultaneously (Figure A1b). The system was designed to keep the 6 axes parallel, thus avoiding stirrers from scratching and damaging the reactor walls.
- Nitrogen inerting, before the reactions, was also implemented via specifically designed inlets on the seals of the glass tubes and a dedicated nitrogen feeding network.

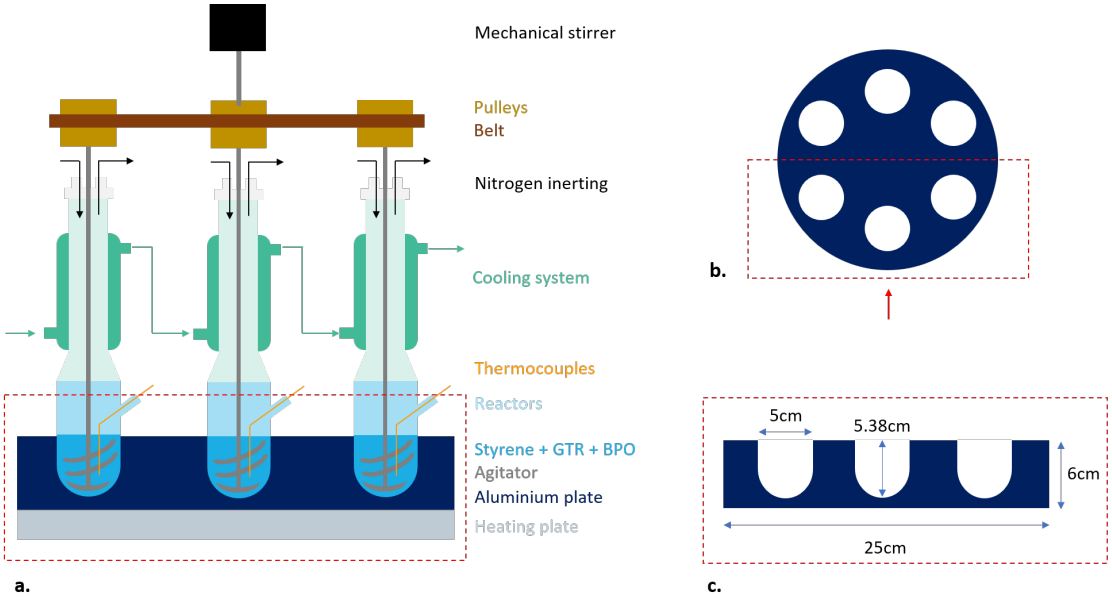


Figure 1. (**a**) Side view of the experimental medium-sized system. (**b**) Top view of the aluminum plate. (**c**) Side view of the aluminum plate from the red arrow.

All reactors were first filled with the desired amounts of GTR. The respective mixtures of monomer and initiator were preliminary prepared in beakers and then introduced in the corresponding reactors directly onto the GTR. An initial blend with the mechanical agitator was performed manually to ensure a good distribution of the mixture of monomer and initiator within GTR particles. The experimental system was then mounted as shown in Figure 1a. Reactors were sealed and purged with nitrogen while initiating the cooling system and the mechanical agitation. The heating plate was turned on only once the cooling thermostat had reached its set-point temperature. During the polymerization, a temperature measurement was taken inside each reactor every 15 min during the first hour, then every hour until the end of the reaction. For these measurements, agitation was momentarily paused. In general, mixtures with low GTR content displayed higher temperature variations during the initial stages of the reaction but, eventually, all reactors reached a steady reaction temperature.

At the end of the polymerization, the reactors were removed from the experimental system and cooled-down in an ice bath to room temperature. The full contents of the reactors were removed, weighed and then transferred in vacuum oven at 40 °C until constant mass. Throughout the different steps of the experimental procedure, special care was taken to measure eventual mass losses of reactants and products (i.e., due to their transfer between recipients or due to evaporation). The uncertainty related to the weighing equipment was also considered. The conversion of styrene during the polymerization reaction $X_{styrene}$ was determined on the basis of an expression similar to Equation (1):

$$X_{styrene} = \frac{m_{i,styrene} - m_{f,styrene} - \Delta}{m_{i,styrene}} \quad (2)$$

where, Δ accounts for the total loss of styrene, from the preparation steps to the final measurement, as detailed in Appendix B. Finally, the measurement uncertainty, due to the equipment used, was assessed on the basis of the principle of error propagation [43] (cf. Appendix B for details).

2.4. Design of Experiments

A wide range of operating conditions, for the different experimental runs, was defined during this study in order to succeed in mapping the widest domain of the feasible experimental space. These were defined in terms of three principal conditions, namely

the reaction temperature, the peroxyde intiator content and the GTR loading, as shown in Table 1. In particular, an effort was made to avoid limiting the amount of GTR to very low values, as it is often the case in other reported studies [12,36], which would also be coherent with an ultimate objective of maximizing the recycling of end-of-life tires [10] in an eventual industrial realization of the system. In this respect, a GTR loading as high as 70% was tested. On the basis of these established ranges, D-optimal designs of experiments were prepared for both experimental setups, considering a maximum reaction time of 8 h per experiment. These are depicted in Figure 2 for both systems.

However, along the experiments and in view of the obtained results, it was decided to slightly modify the initial design by introducing a few additional points to better investigate a specific experimental domain of interest. These points are shown in Figure 2, in orange for the small sized-system and in green and yellow for the medium-sized system at 90 °C and 100 °C, respectively. Inversely, some points were omitted due to extremely low reaction advancement for the adopted reaction temperature and duration (i.e., practically zero styrene conversion) or important temperature overshoot within the first minutes of the reaction. All the realized experiments, along with the measured conversion values, are grouped in Appendix C.

Table 1. Range of operating conditions.

Factors	Min	Max
Temperature	90 °C	110 °C
$m_{i,BPO}/m_{i,styrene}$	3%	8%
$m_{i,GTR}/(m_{i,GTR}+m_{i,styrene})$	10%	70%

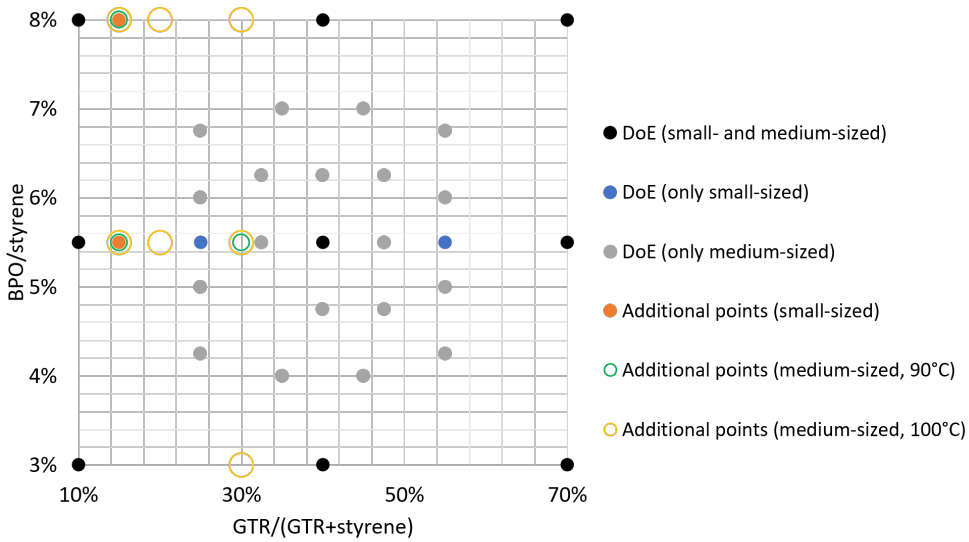


Figure 2. Design of experiments for small- and medium-sized systems.

3. ML Modeling

3.1. ML Algorithms

Data-driven modeling is based on the premise that patterns and trends, which are otherwise difficult to identify or extract on the basis of knowledge and/or observation, can be identified within data coming from a system or process. Among the most common data-driven approaches that are encountered in the modeling of physicochemical systems are response surface methodology (i.e., as part of design of experiments) and ML methods, the latter being particularly adapted to highly complex or multidimensional problems [35]. In the present work, ten popular supervised ML regression models were initially screened on the produced experimental data of the grafting polymerization system to identify the ones

that were more fitted to the specific problem characteristics. Indeed, each ML technique can be more or less suitable to a modeling problem, depending on different factors such as the quantity and nature of the data and the form of the underlying sought pattern. During this initial screening, the parameters of the tested models were fitted to match the experimental data, via the model training step, but without any further optimization of the hyperparameters (HPs) of these models. The tested models belong to different categories, including linear (linear regression (LR) and two of its regularized counterparts, namely ridge and lasso) and non-linear ones (support vector regression (SVR), Gaussian processes (GP) and multilayer perceptrons (MLP)), with extended application in polymer science [34]. In addition, similarity-based (k-nearest neighbors (kNN)), decision trees (DT) and ensemble (random forest (RF) for bagging and gradient boosting (GB) for boosting) methods were also tested.

At a second stage, the four best-performing ML models, namely GB, RF, MLP and SVR, were subjected to HP optimization, in an attempt to further improve their performance. Note that, it is extremely difficult to select a priori the best suited ML technique for a specific problem, mainly due to the plethora of existing methods and to the non-deterministic character of the outcome of the final combination of the model setup, training and HP optimization. This is the reason why this selection is not really justified in most reported studies. However, besides attempting a screening of different techniques and selecting the best performing ones, as is the case here, it is also useful to attempt to identify the characteristics of the best performing models that make them suitable to the problem in hand. This can provide valuable conclusions and serve as future guidance in similar problems. Accordingly, the basic principles of these four techniques are briefly explained below and will serve in the later discussion of the results and the conclusions of the study. Moreover, more detailed descriptions and comparative studies are also reported in the relevant literature [44]. Finally, note that all models were developed in Python, using the Scikit Learn library.

GB and RF are ensemble ML methods as they consist of combining the outputs of a large number of prediction models to obtain an improved ensemble prediction. The difference between GB and RF lies in the way the outputs are combined as well as the construction of the constitutive prediction models. More specifically, they, respectively, belong to boosting and bagging (or bootstrap aggregation) categories. In boosting, simple weak estimators (in underfitting) are combined sequentially in a way that at each iteration, a new weak learner is trained by considering the errors of the previous one (i.e., each new learner tries to correct the errors made by its predecessor), thus reducing underfitting/bias. In GB, the considered learners are DT. The principal advantages of GB are its capacity to model complex non-linear relationships, its customization possibilities and its generalization ability. However, it suffers from large computation cost. The main HPs of GB are generally the number of estimators (or learners) and the learning rate. Other HPs such as the fraction of features and samples considered in each tree can also introduce some randomness which has been proven to ameliorate generalization performance and calculation time. More details can be found in [45–49].

Inversely, in bagging, several well performing prediction models (in overfitting) are trained in parallel on a bootstrap sample of the data set (i.e., a subset of the train data set that is randomly drawn to train an individual prediction model; this subset is then replaced so that all subsets are drawn from the same data distribution). The individual predictions are then combined to give one final prediction (majority vote if classification problem, average value if regression problem) which reduces the overfitting/variance of the individual models. In RF, the prediction models are DT. However, the main particularity of RF method lies in that each DT is build only on a random subset of features. The random subsets of data and features reduce the possible correlations between trees, creating a diversity of individual prediction models (specific to particular conditions), and thus resulting in better accuracy and stability. Moreover, as the number of trees grows, the generalization error converges which makes RF particularly robust against overfitting.

Other advantages of RF are its low sensitivity to HP values and its capacity to evaluate the importance of input features. As RF is based on DT, it also benefits from the latter's simplicity and interpretability. More details about RF method can be found in [44,50–54]. The main HPs of RF are the number of trees and the number of random input features and samples considered for each tree.

MLP, which are a specific type of artificial neural networks (ANN), consist of several layers containing the so-called neurons: an input layer, containing the inputs of the problem, an output layer, containing the outputs of the problem, and one or several intermediate hidden layers. The neurons present in the different layers are interconnected and associated with a weight. The information exchange is made possible thanks to the neurons which transform their input (i.e., the sum of all the inputs arriving from the previous layer multiplied by their corresponding weights plus a bias term) to an output by means of an activation function. Weights are iteratively adjusted during the learning process by minimizing the error between the predicted and the true outputs. A great number of HPs need to be set (and eventually optimized) in MLP, such as the number of hidden layers, the number of neurons in each hidden layer, the activation function, the learning rate and the method to adjust the weights, which renders this step especially time-consuming. On the other hand, MLP presents a powerful capacity to model complex problems as it can approximate any linear or nonlinear mathematical function. On the counterpart, the training of a MLP presents several local minima and is prone to overfitting and lack of generalization, as the complexity and the size of the network architecture increases, especially for small data sets. Furthermore, the structure of the MLP itself makes it harder to interpret compared to RF. More details about ANN and MLP can be found in [55–62].

SVR is the counterpart of support vector machines (SVM) classification algorithm for regression problems and is widely used due to its good generalization ability, high prediction accuracy and robustness during the training process (convex problem), even in presence of small, high dimensional and noisy data sets [63–66]. In SVM, the goal is to find the best hyperplane that will separate the data correctly into classes, which is performed by maximizing the margin (distance between hyperplane and closest point, also called support vector) in both sides of the hyperplane and the number of data classified correctly. When data is not linearly separable, kernel functions can be employed to re-describe them in another higher dimensional space where they will be linearly separable, thus improving classification performance. More details can be found in [63,64,67–70]. The principle of SVR is also based on finding the best hyperplane. However, the goal here is to have the maximum number of points located inside the narrowest margin (or epsilon tube), the latter containing the hyperplane which represents the regression function. The main HPs are the regularization parameter C (to penalize the highest deviations outside the margin), epsilon (radius of the epsilon tube) and the kernel. Other HPs, more specific to the kernel, can also be optimized.

3.2. ML Procedure

In any data-driven modeling study, the data pretreatment represents the most crucial and time-consuming step, as the quality of data (besides quantity) will largely determine the performance of the model (i.e., widely known as "garbage in, garbage out" principle). In this work, data pretreatment consisted essentially in keeping the most reliable experimental measurements (i.e., eliminating the ones in which important losses or measurement errors were observed). The final data set, that was used for the training of all ML models, is composed of a total of 69 experiments, as shown in Table C1. These were shuffled and partitioned into train and test sets according to a 5-fold cross-validation procedure (Figure 3a). The latter enabled to evaluate the variability of the performance subject to partitioning, while ensuring that each sample was present in one of the test sets. As part of the data pretreatment process, the considered inputs need also to be standardized (i.e., rescaled to present a common mean and standard deviation) as means to ensuring that their eventual difference in magnitude will not bias the model. In the present work, a mean

value of 0 and a standard deviation of 1 were used for the normalization, while this was applied solely on the train set to avoid data leakage (i.e., assure the complete independence of the train and test sets).

For the HP optimization step, a grid search cross validation method was employed to exhaustively test all possible combinations. This was motivated also by the relatively low number of models/HPs to be tested. In a different case, other HP optimization strategies, such as genetic algorithms or random search, could be employed, depending on the problem characteristics and the type of ML algorithms [71]. To fine-tune the boundaries and levels of the considered HPs, a prior manual investigation of the impact of each individual HP on the model performance was initially carried out. The different types of HPs for each model, along with their respective optimization domain, are detailed in Table 2.

A nested cross-validation scheme was adopted in order to limit the introduction of bias in combined HP tuning and model selection steps. More specifically, this procedure consists of two nested cross-validation schemes, namely an external one based on a train/test split of the whole data set and an internal one based on a train/validation split of each training set of the outer loop. The outer loop is used for model training (i.e., estimation of its parameters) and model selection, while the inner one aims in HP optimization. In this work, 5-fold outer and inner cross-validations were employed (Figure 3b). Even if this procedure is more time-consuming than using a common k-fold for both model training/selection and HP optimization, it produces more robust and unbiased performance estimates against overfitting and is highly suitable for small data sets [72,73].

Among the different metrics that are typically employed to evaluate the performance of ML models, the root mean squared error (RMSE), mean absolute error (MAE), determination coefficient (R^2), max and median error, are the most commonly encountered ones. In this work, RMSE was selected as the principal model evaluation metric as it penalizes specifically the largest errors, which are to be avoided if the model has to be applied for industrial use. However, in the results section, R^2 and MAE values are also reported in order to provide a more complete overview of the performance of the developed ML models.

Table 2. HPs tested for GB, RF, MLP and SVR models.

ML Algorithm	HPs	Values
GB	n_estimators	[50, 100, 150]
	learning_rate	[0.1]
	max_features	['sqrt', 'log2']
	min_samples_leaf	[1, 5, 10, 15]
	subsample	[1/5, 2/5, 3/5, 4/5, 1]
RF	n_estimators	[50, 100, 150]
	max_features	['sqrt', 'log2']
	max_samples	[None] (no bootstrap: all train samples)
MLP	activation	['relu']
	hidden_layer_sizes	1 hidden layer: [(i)] with i = 25, 50, 75, 100, 125, 150 2 hidden layers: [(i, j)] with i, j = 5, 10, 15, 20, 25, 30, 50
	solver	['lbfgs', 'adam']
	learning_rate_init	[0.001, 0.005, 0.01, 0.04, 0.07]
	max_iter	[800]
SVR	kernel	['rbf']
	C	[0.5, 1, 1.5, 2, 3, 4, 5, 6, 8, 10]
	epsilon	[0.01, 0.05, 0.1]

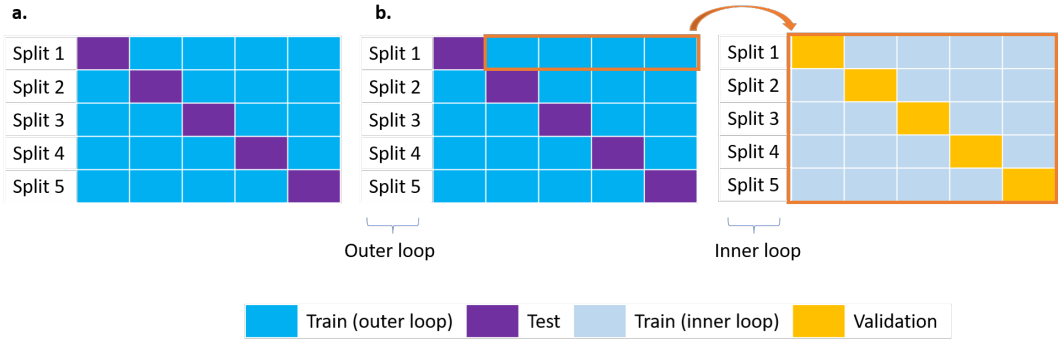


Figure 3. Data partitioning procedures: (**a**) 5-fold cross-validation procedure for ML models screening without HP optimization. (**b**) Nested cross-validation with 5-fold outer and 5-fold inner cross-validation procedures for combined model selection and HP optimization.

4. Results and Discussion

4.1. Experimental Results

All the results for the small-sized and medium-sized systems are presented in detail, respectively, in Tables C1 and C2 and are also publicly accessible in electronic format (cf. Data Availability Statement). In the following, the behavior of the two systems in terms of temperature control, polymerization time and effect of GTR and BPO is discussed, on the basis of the obtained results. In all the presented graphs, error bars represent standard deviation (i.e., whenever an experiment has been repeated several times). The average absolute uncertainty in the determination of monomer conversion was estimated (cf. Appendix C) to be equal to an average of 0.13% wt and 0.29% wt for the small-sized and the medium-sized systems, respectively. This was primarily based on the uncertainty of the equipment used (i.e., $\pm$0.1 mg of the precision balance). At the same time, the average standard deviation of three measurement repetitions for a total of seven samples was found to be equal to 0.3 mg. Finally, other sources of introduction of random error to the measurements, such as the balance drift, influenced by air stream, temperature or humidity, were not quantified.

During the experiments, the temperature inside the reactors was investigated to ensure the reliability of the data for latter modeling. In the small-sized system, a thermocouple was kept inside three reference tubes (corresponding to GTR/(GTR + styrene) contents of 10%, 40% and 70% wt, the ratio BPO/styrene being set at 5.5% wt) during the whole reaction for the three temperatures of interest. An example of the recorded temperature evolution, for 90 °C, is shown in Figure 4 and in Table 3. Despite an initial overshoot of the reaction temperature, which is more pronounced for mixtures with lower GTR content, the set-point temperature was reached in the first 10 min of reaction, within a margin of $\pm$2 °C in all cases. In the medium-sized system, it was impossible to follow the evolution of the internal temperature on a permanent basis due to the form of the mechanical agitator that covered the reactor radially and throughout the complete mixture height. Accordingly, at specific time-intervals, the agitation was stopped promptly and thermocouples were introduced into the reactors through the lateral orifice for temperature measurement. In this case, the temperature overshoot was significantly more pronounced, reaching even 56 °C at certain conditions (see Table C2 for all temperature measurement results). Note that, the measured temperature was also affected by the positioning of the thermocouple inside the reactor, a phenomenon that was not observed in the small-sized system. The highest temperature values were observed below the agitator, near the bottom of the reactor, which are the ones that are reported here. These observations validate the ones made during a previous work [28] in a larger scale reactor and are evidence of the scale-up difficulty of the system. They are also indicative of the necessity to accompany published experimental measurements in similar systems with details about the achieved temperature homogeneity and the eventually observed heat-transfer limitations. In view of these elements, in the

present work, 110 °C experiments were limited for the medium-sized system while in the subsequent ML modeling treatment, only small-sized system data were considered as isothermal experiments.

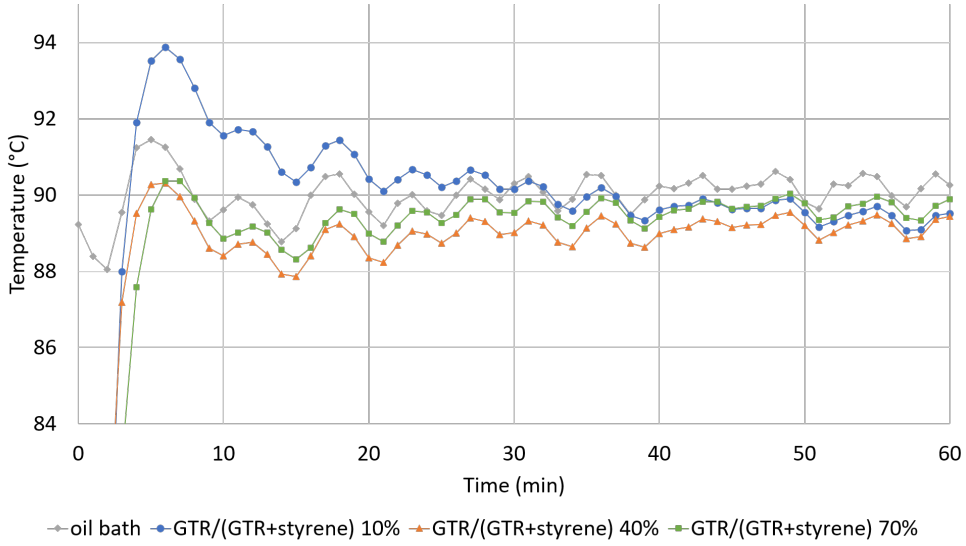


Figure 4. Polymerization temperature for small-sized system for different GTR/(GTR + styrene) ratios (% wt) (90 °C, BPO/styrene 5.5% wt).

Table 3. Temperature control in the small-sized system (BPO/styrene = 5.5% wt).

T:	**90 °C**			**100 °C**			**110 °C**		
GTR/(GTR + Styrene):	**10%**	**40%**	**70%**	**10%**	**40%**	**70%**	**10%**	**40%**	**70%**
Mean T after 1 h 30	89	89	89	99	99	100	106	106	109
Min T after 1 h 30	88	87	88	98	98	98	104	105	108
Max T after 1 h 30	90	90	91	101	101	101	108	108	111
t (min) to reach T ± 2 °C	3	3	4	2	3	3	2	-	5

Another parameter of interest, for this polymerization system, is the reaction time that, depending on the conditions, may extend up to >30 h until complete styrene or BPO consumption [12,36]. In the present work, the maximum reaction time was limited to 8 h due to technical constraints. However, the experiments carried out in the small-sized system, under the different reaction conditions set in the experimental design, were performed also for shorter duration, namely for 2 h and 5 h (see Table C1). This allowed monitoring the temporal evolution of the system and including time as a factor in the subsequent ML modeling study.

The evolution of styrene conversion, with respect to the mixture composition in GTR and BPO, is presented in Figures 5 and 6, for a reaction temperature of 90 °C and 100 °C, respectively. In these figures, continuous curves correspond to the small-sized system (denoted as S) while dashed curves are for the medium-sized system (denoted as M). The overall trend, observed for both systems and in both temperatures, is completely consistent with previous observations [27–29] where an increase in the amount of GTR in the system displayed a negative effect on monomer conversion. At the same time, a rather expected positive impact of the amount of BPO on the final styrene conversion is also observed. The inhibiting effect of GTR has been attributed to the reaction between the dissociated benzoyl acid radicals and the unpaired electrons on the surface of carbon black, creating new active sites which are either inert or have lower reactivity than the BPO primary radicals, thus limiting styrene conversion. Another possible mechanism, acting synergetically towards the same direction, could be the redox reaction between BPO and the oxide groups of the

carbon black surface, which would be favored at a lower reaction temperatures due to its lower activation energy than the thermal decomposition of BPO [27].

It is also worth noting that the effect of GTR content on the value of monomer conversion is not linear, with respect to the GTR content, but is more pronounced in the range of 10–40% wt loadings. For example, at 90 °C (i.e., Figure 5), the final monomer conversion after 8 h of reaction drops from around 80% to only 10% when passing from 10% to 40% wt in GTR content. However, half of this decrease is already evident at 15% GTR content. This phenomenon had already been identified in previous studies [27,29] and a critical GTR concentration had been considered, in the range between 10% and 50% wt of GTR content, beyond which the observed inhibiting effects are gradually attenuated.

The same trend is observed in both small-sized and medium-sized systems. At the same time, there is a slight disagreement between the results of the two systems, under similar reaction conditions, with the small-sized system displaying higher styrene conversions at lower GTR/(GTR + styrene) ratios (i.e., 10% and 15%) than the medium sized one. This tendency gradually inverses as the GTR ratio increases (i.e., up to 70% wt of GTR). This difference could be attributed to the aforementioned unavoidable mass losses that occur during the different steps in the medium-sized system, especially during the transfer of the reactor content to the recipients for mass measurement and evaporation in the vacuum oven (e.g., non-recovered material from the reactor walls and agitator). Although an effort was made to meticulously monitor these losses and account for them during the calculations (cf. Δ_2 in Equations (2) and (B.1)), it was assumed that the total losses in mass is proportionally distributed among styrene and GTR, according to their initially introduced amount in the reactor. In this sense, for low GTR/(GTR + styrene) ratios, a relatively higher percentage of the measured mass loss will be attributed to styrene, thus leading to the calculation of a higher value for monomer conversion than in a case where the same mass loss would correspond to a higher GTR content. A different consideration of the distribution of losses (e.g., an equal distribution between GTR and styrene) would have artificially decreased this observed deviation between the two systems. In any case, it remains technically impossible to verify this assumption.

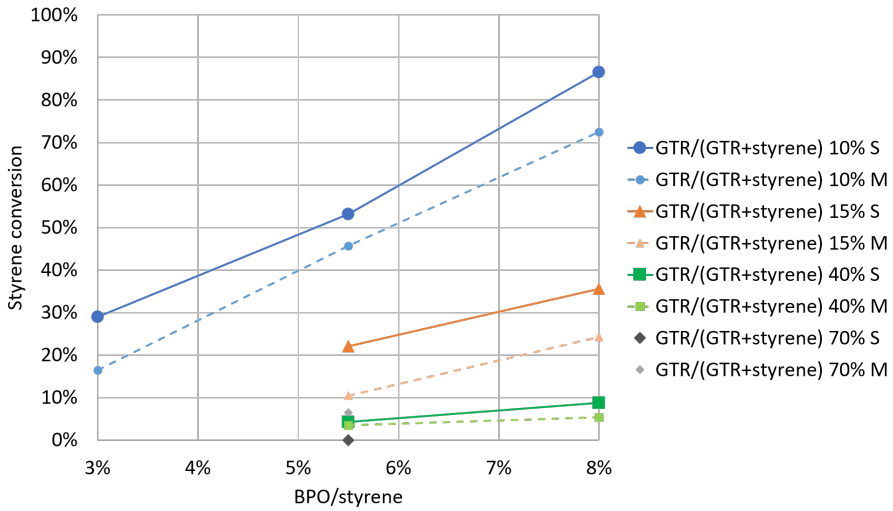


Figure 5. Comparison of small- (S) and medium- (M) sized systems at 90 °C.

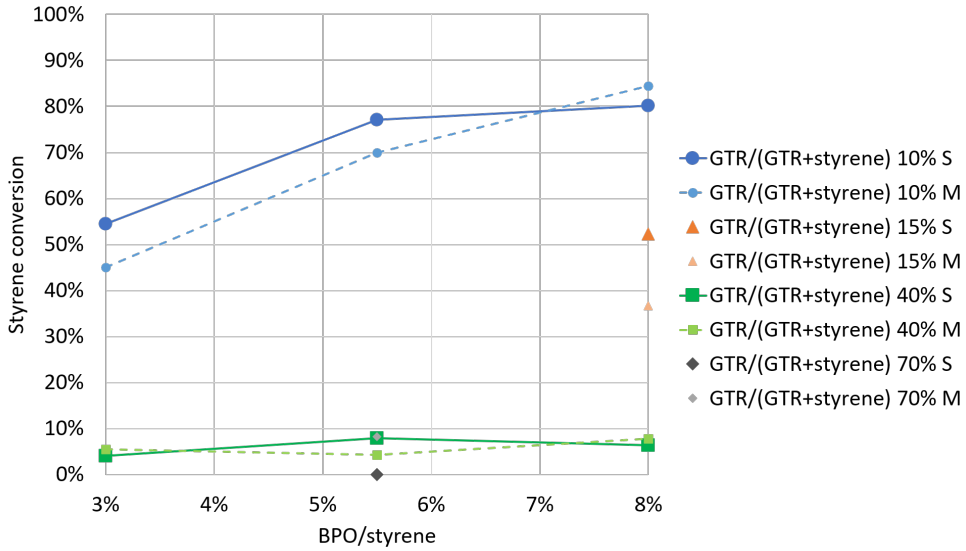


Figure 6. Comparison of small- (S) and medium- (M) sized systems at 100 °C.

Figures 7 and 8 depict the combined effect of GTR content and reaction temperature, for a constant amount of initial BPO concentration, on the measured styrene conversion. Besides the negative effect of GTR content, it is seen that the reaction temperature displays an overall positive effect by increasing styrene conversion. It is also observed that the effect of temperature becomes less profound at higher GTR loadings. These observations are completely inline with the previously reported measurements [27] as well as with the proposed hypothesis of redox reaction between BPO and the oxide groups of the carbon black surface that would be favored at lower reaction temperatures.

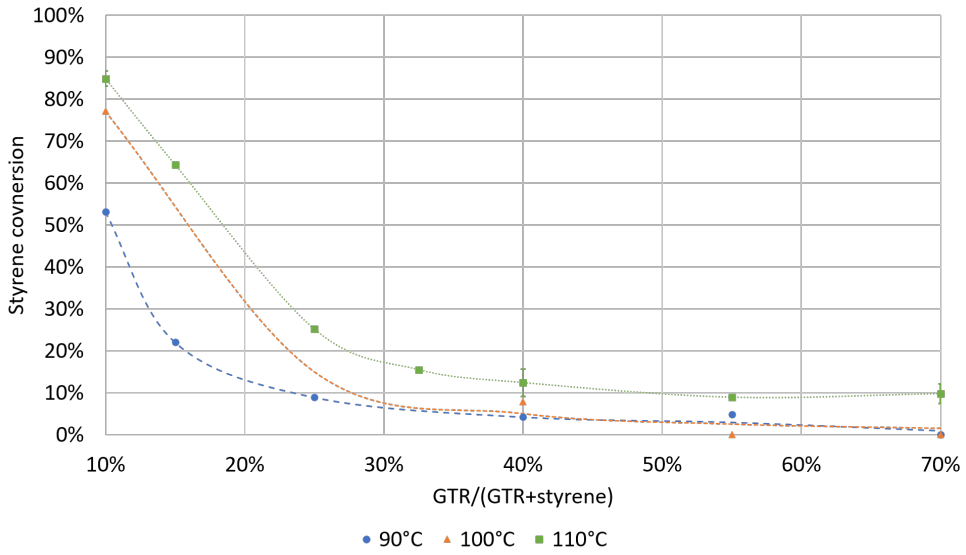


Figure 7. Effect of GTR/(GTR + styrene) and temperature for small-sized system and BPO/styrene = 5.5% (dashed curves correspond to trend lines).

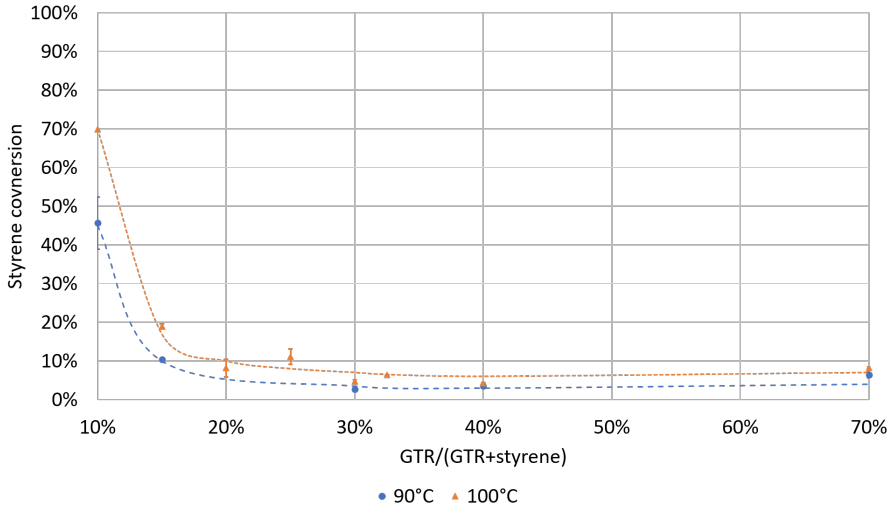


Figure 8. Effect of GTR/(GTR + styrene) and temperature for medium-sized system and BPO/styrene = 5.5% (dashed curves correspond to trend lines).

4.2. ML Results

Following the pretreatment of the measured experimental data, as explained in Section 3, 10 different supervised-learning regression algorithms were screened and compared, prior to any further optimization of their HPs. In Figure 9, the parity plots of the split corresponding to the lowest RMSE, for the test data set, are presented for each model. In addition, comparative bar plots between all tested models, on the basis of the adopted performance criteria (i.e., R^2, RMSE, MAE), are depicted in Figures 10 and 11 for the train and test data sets, respectively. The error bars correspond to the standard deviation, calculated on the basis of the 5-fold learning procedure. The exact values of all the metrics are also provided in Table D1. The best performing models, identified in terms of their test data set performance metrics were found to be the ensemble models (GB and RF), DT, MLP and SVR. The corresponding test R^2 values for these models varied between 0.927 ± 0.032 and 0.762 ± 0.066.

These models, identified as the top-performing ones in this case, are among the most-commonly employed in similar reported studies [35]. By their construction, ensemble models can generalize very well to new data as they are composed of several models, each trained under specific data conditions. DT models also perform particularly well, but typically show lower performance than ensemble models as they can be considered as a sub-case of RF. MLP networks are notorious for their capacity in modeling highly non-linear response surfaces but suffer from low generalization and a need for extensive data-sets. On the other hand, SVR presents a more robust training performance and perform well with limited data. It would thus seem as these models displayed better performance than the rest due to a combination of characteristics of the problem that would partially match their specific advantages. Accordingly, GB, RF, MLP and SVR were selected for the HP optimization phase, in order to investigate whether further significant improvement could be attained in their performance, differentiating one (or more) of them as a clear choice for the modeling of this system. Note that, the selection of these models was not based strictly on their performance but also on a choice to further test models that are structurally different (i.e., ensemble models vs. SVR and ANN) and quite popular in such applications. Note also that some of them, such as SVR and MLP, are typically considered to be more sensitive to their HP values than others (e.g., RF). Finally, the low values of standard deviation for the selected models are indicative of their robustness against the train/test splits, which allows generalizing the conclusions about their applicability with higher confidence.

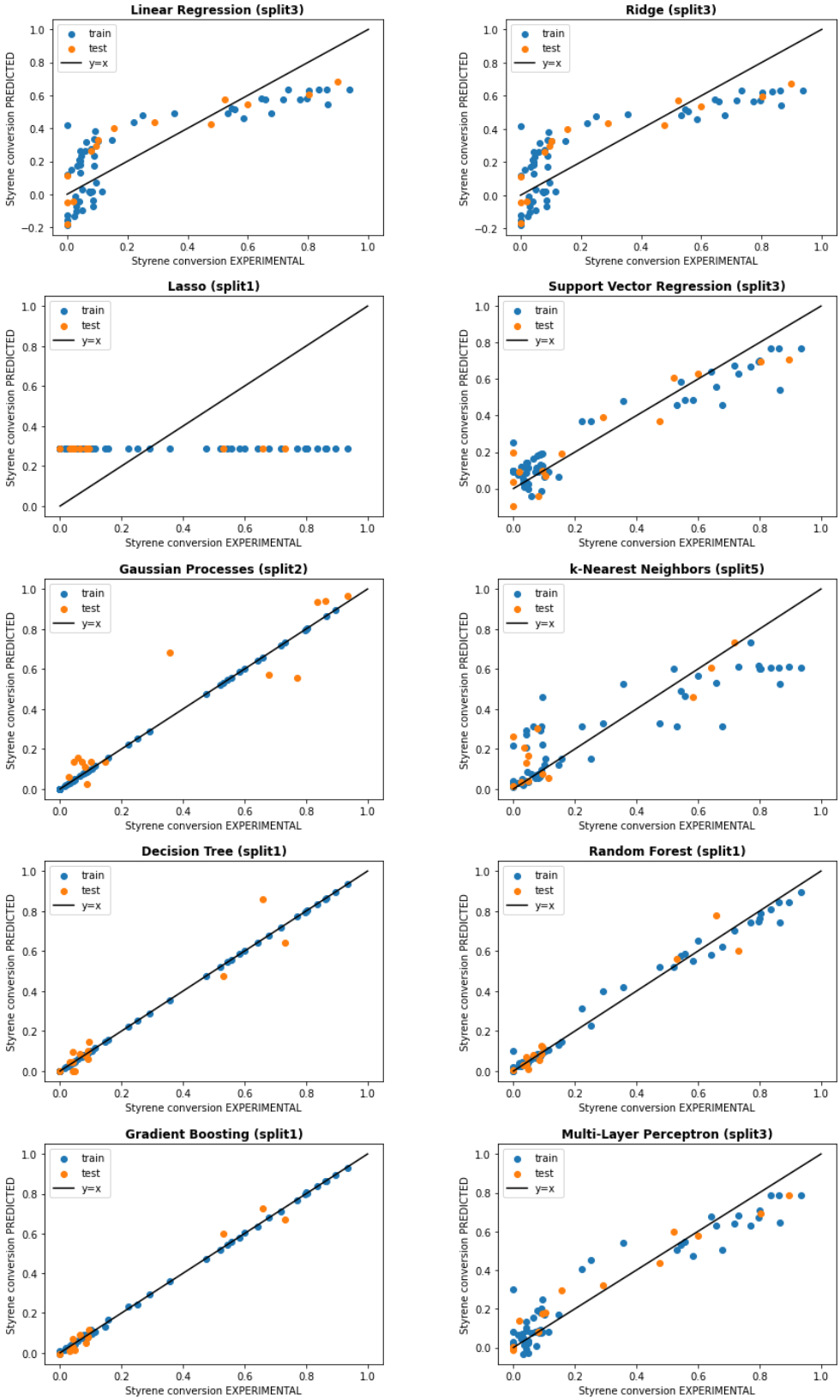


Figure 9. Parity plots of the ten screened models (without HP optimization).

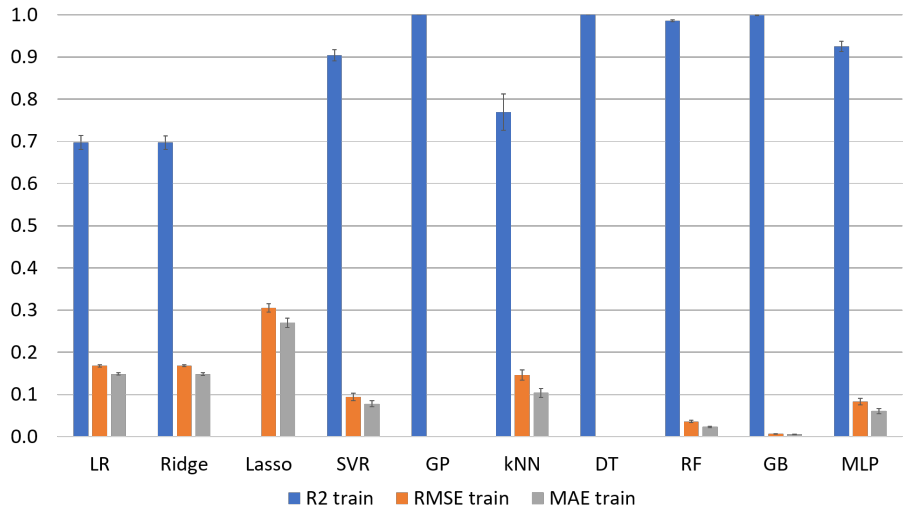


Figure 10. Train performances of the ten screened ML models (without HP optimization).

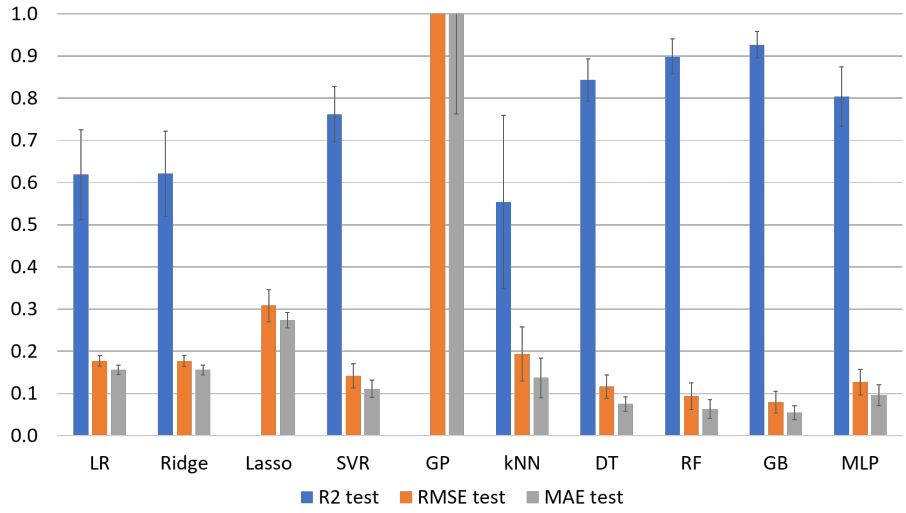


Figure 11. Test performances of the ten screened ML models (without HP optimization).

Concerning the rather poorly performing models, it seems that the experimental data could not be described by linear models and/or by the use of a reduced number of input features (i.e., as is the case in strongly regularized linear models), thus leading to evident underfitting. This becomes obvious in the case of Lasso, which significantly reduces the number of input features (e.g., as opposed to Ridge). An optimization of the regularization parameter in Ridge and Lasso models could probably help to improve the performance results and/or to reduce underfitting but this was not further pursued in this work. On the contrary, GP model performs perfectly on the training data set but fails to generalize in the test set, displaying clear evidence of severe overfitting. Similarly, an optimization of the GP kernel could have been explored as means to improve its performance, just as an optimized value of the number of neighbors might had improved the performance of the kNN model and render it more interesting for this problem.

The four best-performing ML models (GB, RF, MLP and SVR) were subsequently subjected to HP optimization using a "nested cross-validation" procedure, instead of a simple k-fold validation (i.e., as the one applied before HP optimization). The obtained optimized HPs are presented in Table 4, for each of the 5 train/test splits of the 5-fold outer cross-validation, and are completed with the average computation time for one split. Note that certain HPs of some of the models, such as the number of estimators in RF, the hidden

layer sizes in MLP and the regularization parameter C in SVR, vary depending on the split, while the rest are not affected. It is also observed that the train/test split displays a more or less significant effect on the model performance, depending on the method. For example, Figure 12 shows that the test R^2 of SVR with optimized HPs varies from 0.7 to 0.9, depending on the split. However, the overall net improvement of SVR and MLP models, after HP optimization, is also clearly visible in Figure 13, where train and test data are better aligned on the $y = x$ bisector of the parity plots.

Table 4. HPs optimized for each train/test split, for GB, RF, MLP and SVR.

ML Algorithm	HPs	Split 1	Split 2	Split 3	Split 4	Split 5	Time (1 Split)
GB	n_estimators	150	150	150	150	150	3.8 s
	learning_rate	0.1	0.1	0.1	0.1	0.1	
	max_features	'log2'	'log2'	'log2'	'log2'	'log2'	
	min_samples_leaf	5	5	5	1	1	
	subsample	1	1	0.8	0.6	0.4	
RF	n_estimators	50	150	150	50	50	1.2 s
	max_features	'log2'	'log2'	'log2'	'log2'	'log2'	
	max_samples	None	None	None	None	None	
MLP	activation	'relu'	'relu'	'relu'	'relu'	'relu'	27.5 s
	hidden_layer_sizes	(15, 10)	(50, 5)	(5, 5)	(5, 15)	(5, 25)	
	solver	'lbfgs'	'lbfgs'	'lbfgs'	'lbfgs'	'lbfgs'	
	learning_rate_init	0.001	0.001	0.001	0.001	0.001	
	max_iter	800	800	800	800	800	
SVR	kernel	'rbf'	'rbf'	'rbf'	'rbf'	'rbf'	0.7 s
	C	1	4	8	1	2	
	epsilon	0.01	0.01	0.01	0.01	0.01	

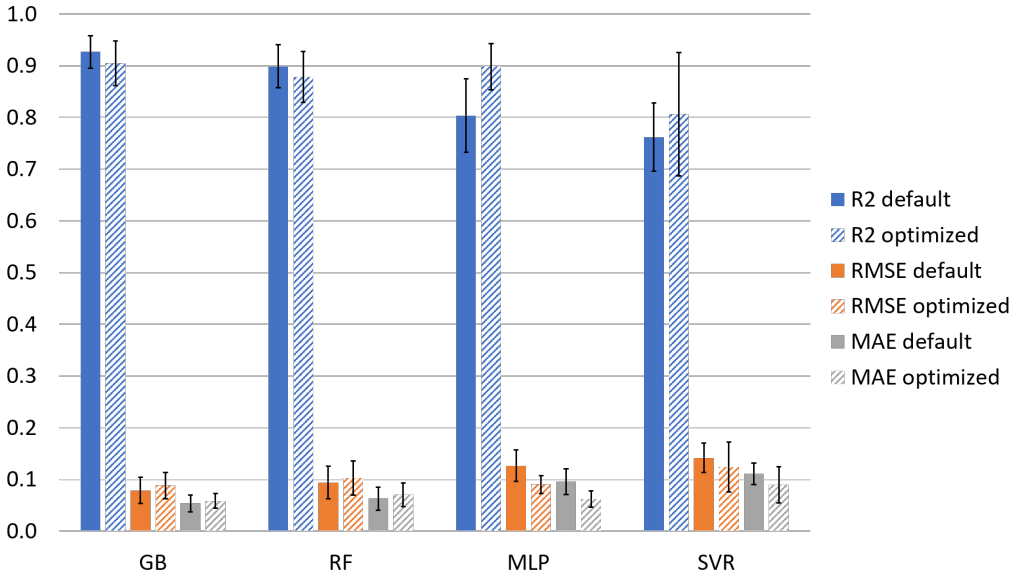


Figure 12. Comparison of test performances of GB, RF, MLP and SVR with default vs. optimized HPs.

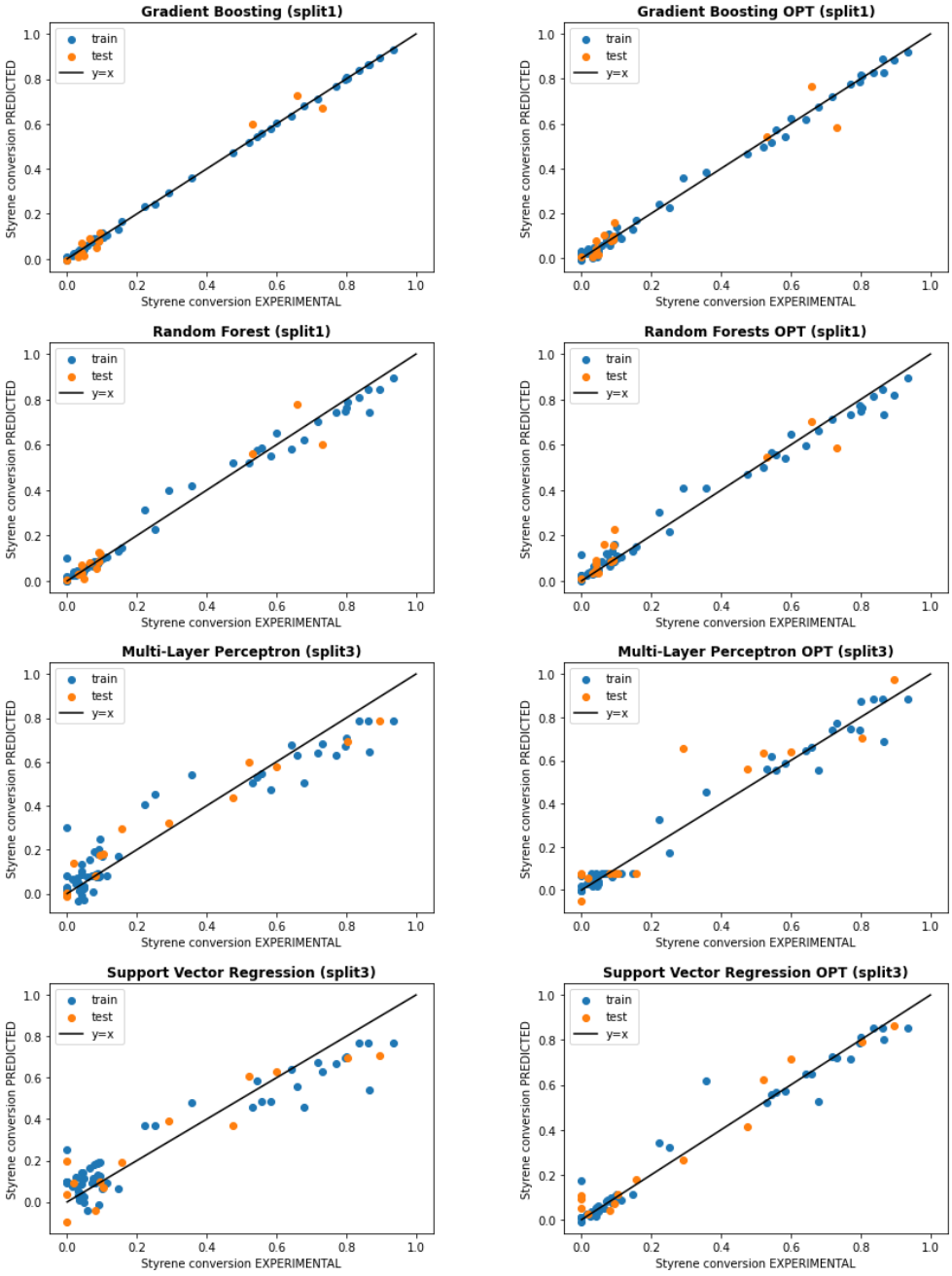


Figure 13. Comparative parity plots (with default HPs on left side, and with optimized HPs on right side).

Once these optimized values were obtained, they were used as part of the 5-fold outer cross-validation, to train the models on train set and evaluate them on test set for each train/test split. The test performances (R^2, RMSE and MAE) with default (i.e., prior to HP optimization) and optimized HPs are compared in Figure 12, where full bars represent the performances with default HPs while hatched ones are for performances with optimized HPs. The error bars represent the standard deviations calculated on the basis of the five different train/test splits. The exact values of all the metrics are also provided in Table D2. Note that the final results are subject to the selected HP optimization method and metric. Accordingly, the selection of a method different than grid search cross validation or the optimization on the basis of another metric than RMSE, would have probably resulted in different values for the optimized HPs and for the model performances. A net improvement is observed in the SVR and MLP model performances on the test data set, whilst GB and RF

display no improvement but rather a slight degradation in their performance. Moreover, as discussed previously, RF models are known for showing low sensitivity to their HPs values. The improvement of SVR and MLP models after HP optimization is also visible in Figure 13, where train and test data are better aligned on the $y = x$ bisector of the parity plots. Finally, note that the test performance of the models displays a higher variation than their respective train performance (cf. Table D2), which can be considered as further justification of the need to employ a k-fold validation procedure rather than evaluating the performance of the models on the basis of any random split.

Several options could be envisioned to further improve the prediction performances of the models. First of all, it is clear that the limited number of samples prevents the models from learning from more data and thus improving their generalization performance. Indeed, as discussed in the introduction of this work, data-driven techniques require sufficient amount and quality of data. Although it is not evident to define *a priori* what "sufficient" means, especially since different methods have different requirements in terms of the amount of data, relative studies have shed light to the interplay that exists between the amount of data, used for the training of ML models, and their precision [74]. Accordingly, in the case of problems of limited data, one option would be to introduce a crude estimation of the target property in the feature space to enhance the accuracy of ML models.

At the same time, it is clear that fundamental experimental studies of physicochemical systems or materials cannot compete, in terms of data availability, with problems related to the fields of computer science, medical records or industrial data analysis. In this sense, it remains crucial that the experimental procedure is rigorous, involving identification and monitoring of all the steps that introduce uncertainties to the measurements. The impact of the experimental uncertainties on ML models performance can be subsequently evaluated and taken into account in the model development phase. For example, the incorporation of known data uncertainty measures on input and/or target data has been previously investigated for ANN, by using least-squares-inspired methods, and resulted in better generalization capacity [75–78]. In the present work, this possibility was exploited via a weighing procedure of the data used in the training steps of SVR, RF and GB by the estimated uncertainties on monomer conversion measurements. In particular, the normalized inverses of the uncertainties were used to weigh the samples in the regression functions (i.e., through the option sample_weight). However, only a marginal improvement was observed solely for SVR, corresponding to an increase in R^2 from 0.806 to 0.817, thus this option was not further pursued. The comparison of the test performances between the default case (all samples have same weight) and the case with weighted samples, after HP optimization, is provided in Figure E1.

Other parameters related to the ML model development procedure, such as the ratio for the partitioning of data into train and test set, the data standardization method and the random data shuffling and model initialization, could also influence the final model performance. The influence of these parameters in the performances of GB, RF, MLP and SVR, prior to HP optimization, was also studied in this work. The results are provided in Figures E2–E4, respectively, for the three previously described parameters. They showed that SVR can be slightly improved by using MinMax scaling instead of standard scaling. At the same time, MLP showed a net degradation of its performance with MinMax and robust scaling while the other two methods did not display any significant difference among the three scaling techniques. The partitioning ratio 90/10 for train and test sets, corresponding to 10-fold cross-validation, despite improving the average test performance was accompanied by higher standard deviations. Indeed, more data was available to train the models in this case but, at the same time, the model was evaluated on a very small test set which resulted in a broader error distribution. Finally, the random state, which controls the sequence of the pseudo-random number generator, was also found to display an effect on the model performance. As discussed previously, this effect can only be counterbalanced by the implementation of a k-fold cross validation procedure for the model training and HP optimization steps. However, it is important to remain aware of this random fluctuation

of the model performance, especially given the limited amount of available data. Finally, another option for potentially improving the model performances would be by integrating their data-driven character with existing knowledge on the system. One way to achieve this would be to couple them with the previously developed mechanistic model [29] in an attempt to exploit simultaneously the advantages of each one of the two approaches in a unique hybrid model. This remains a perspective for future work.

5. Conclusions

The study of the kinetics of the styrene–GTR radical graft polymerization represents a great challenge due to the complexity of the prevailing kinetic mechanisms as well as the nature of GTR. In addition, the existing knowledge or controlling capacity of these aspects remains limited. In this respect, a rigorous experimental study and a ML modeling approach were combined in this work in an attempt to overcome these limitations and relate operating conditions with a key performance index, namely styrene conversion.

The particularity of the adopted experimental approach lied in the attention paid in ensuring data reliability and quantity, which are crucial in any ML approach. For this purpose, two laboratory-scale experimental setups were implemented to evaluate the reactive medium thermal and compositional homogeneity and investigate scale-up effects. The experimental procedure for both systems was designed so that data can be produced rapidly, while increasing data reliability by eliminating the uncertainty related to sampling for analyses. At the same time, all the potential sources of uncertainty, such as the mass loss along the different steps of the process and the precision of the experimental equipment, were carefully identified and monitored. The analysis of the results on both experimental scales revealed that the effects of GTR, BPO and temperature on styrene conversion were coherent with previous studies. However, the heterogeneous nature of the system created difficulties in the mixing and the estimation of the losses for the medium-sized system, which should not be neglected in view of an eventual further scaling-up.

The most reliable experimental data, containing 69 experiments from the small-sized system, was subsequently subjected to the screening of diverse supervised-learning regression ML models and the optimization of their HPs to identify the best-performing ones. These were identified to be GB, MLP and RF with test R^2 of, respectively, 0.91 ± 0.04, 0.90 ± 0.04 and 0.89 ± 0.05. In view of the small available data set, a nested cross-validation scheme was employed to account for variance in both validation and test sets, and the results revealed the impact of the random data partitioning on the performance of the developed models. Finally, the effect of additional parameters, such as the scaling method and the number of folds, as well as the integration of the experimental uncertainties in the learning procedure, were exploited as means to improve the predictions of the models.

From the perspective of upscaling the studied system towards industrial application, further research on how to enhance the homogeneity of temperature and composition is more than necessary, especially at higher GTR loadings. The latter are indeed necessary to increase the recycling of used tires. To limit the costs of the process, the use of commercially available reagents and GTR without further purification was preferred in this work. However, the presence of additives, stabilizers or impurities affects the polymerization course, and the variable GTR composition represents a major part of this added uncertainty. This complexity encourages further use of ML tools, ideally in combination with already developed phenomenological models in hybrid approaches, to quickly map and, if possible, understand the relationship between operating conditions and final product properties at different scales. Developing ML models compatible with small data sets and capable of integrating/predicting uncertainties is therefore an interesting direction for future work, as well as the extension to other molecular properties. For the latter, experimental procedures enabling the collection of sufficient data of good quality are therefore required. Finally, an effort to share experimental data and accompany each experimental result with precise and detailed conditions is crucial, as demonstrated in this work.

Author Contributions: Conceptualization, all; methodology, C.T. and D.M.; literature review, C.T.; experimental study, C.T.; design and implementation of the medium-sized system, R.L. and S.H.; data curation and modeling, C.T.; writing—original draft preparation, C.T.; writing—review and editing, C.T. and D.M.; materials, S.H.; supervision, D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by MESRI (Ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation), France.

Data Availability Statement: Experimental data and ML code are available at https://github.com/cindyytrinhh/Machine-Learning-modeling-of-styrene-Ground-Tire-Rubber-radical-graft-polymerization, published on 20 February 2023.

Acknowledgments: The authors acknowledge the "Service Etudes et Réalisation Mécanique" of LRGP as well as the "Service de soufflage du verre" of University of Lorraine who, respectively, designed and built the medium-sized system and the glass parts. The authors also acknowledge the following masters students who contributed in setting up and collecting experimental data: Redouanne Chenna, Justine Guerry, Bahia Lamari, Betty Soulet and Jérémy Lanoizelée.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial neural networks
BPO	Benzoyl peroxide
DT	Decision trees
GB	Gradient boosting
GP	Gaussian processes
GTR	Ground tire rubber
HP	Hyperparameter
kNN	k-nearest neighbors
LR	Linear regression
MAE	Mean absolute error
ML	Machine learning
MLP	Multilayer perceptrons
PS	Polystyrene
RF	Random forest
RMSE	Root mean squared error
SVM	Support vector machines
SVR	Support vector regression

Appendix A. Additional Photos of the Experimental Medium-Sized System

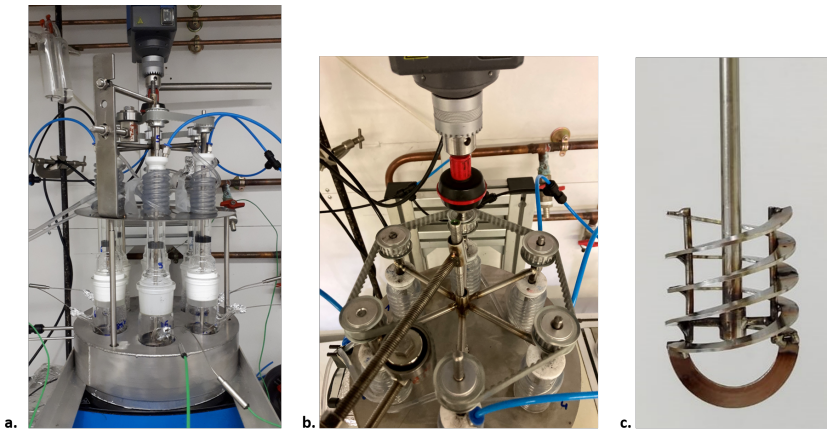


Figure A1. Medium-sized experimental system. (**a**) Whole system. (**b**) Pulleys-belt. (**c**) Mechanical agitator.

Appendix B. Evaluation of the Experimental Uncertainties

During the different steps of experimental procedure, especially in the medium-sized setup, several losses of reactants take place. These need to be identified and quantified for the correct calculation of styrene conversion (i.e., denoted as Δ in Equation (2)). These losses are identified as follows:

$$\Delta = \sum_{i=1}^{3} \Delta_i \tag{B.1}$$

where:

- Δ_1 is the loss of styrene during polymerization reaction due to evaporation;
- Δ_2 is the loss of styrene during the transfer of the reactor content to the gravimetry cup at the end of the polymerization;
- Δ_3 is the styrene remaining in the glass tube at the end of the polymerization reaction.

All the above losses were identified by meticulous weighing of the used equipment before and after each individual step, using a Mettler Toledo XP504 precision balance. During these measurements, the ratio of styrene to GTR was considered to be identical and constant, as defined in the recipe of each experiment and measured during the initial material introduction step.

According to [43], the uncertainty of a function $f = f(x_1, x_2, \ldots, x_n)$ can be expressed in terms of the uncertainties u_{x_i} of each one of its variables x_i and partial derivatives $\frac{\partial f}{\partial x_i}$ (for $i = 1 \ldots n$), as:

$$u_f = \sqrt{\sum_{i}^{n} \left[\frac{\partial f}{\partial x_i} u_{x_i} \right]^2} \tag{B.2}$$

The application of Equation (B.2) to $X_{styrene}$ and Δ, defined in Equations (2) and (B.1), respectively, results in:

$$\begin{aligned} u_{X_{styrene}} &= \sqrt{\left[\frac{\partial X_{styrene}}{\partial m_{i,styrene}} u_{m_{i,styrene}} \right]^2 + \left[\frac{\partial X_{styrene}}{\partial m_{f,styrene}} u_{m_{f,styrene}} \right]^2 + \left[\frac{\partial X_{styrene}}{\partial \Delta} u_{\Delta} \right]^2} \\ &= \sqrt{\left[\frac{(m_{f,styrene} + \Delta) * u_{m_{i,styrene}}}{m_{i,styrene}^2} \right]^2 + \left[\frac{u_{m_{f,styrene}}}{m_{i,styrene}} \right]^2 + \left[\frac{u_{\Delta}}{m_{i,styrene}} \right]^2} \\ &= \frac{1}{m_{i,styrene}} \sqrt{\left[\frac{(m_{f,styrene} + \Delta) * u_{m_{i,styrene}}}{m_{i,styrene}} \right]^2 + u_{m_{f,styrene}}^2 + u_{\Delta}^2} \end{aligned} \tag{B.3}$$

$$u_{\Delta} = \sqrt{u_{\Delta_1}^2 + u_{\Delta_2}^2 + u_{\Delta_3}^2} \tag{B.4}$$

Appendix C. Detailed Experimental Data for the Small- and Medium-Sized Systems

Table C1. Experimental results for the small-sized system.

N°	Time, h	Temperature theo., °C	Temperature exp., °C	GTR/(GTR+Styrene) theo., %wt	GTR/(GTR+Styrene) exp., %wt	BPO/Styrene theo., %wt	BPO/Styrene exp., %wt	Conversion, %wt	Balance-Related uncertainty, %wt
1	2	90.0	89.0	10.00%	10.23%	5.50%	5.49%	47.57%	0.13%
2	2	90.0	89.0	40.00%	40.35%	5.50%	5.49%	0.00%	0.13%
3	2	90.0	89.0	70.00%	69.94%	5.50%	5.49%	0.00%	0.07%
4	5	90.0	89.0	10.00%	9.79%	5.50%	5.49%	58.41%	0.13%
5	5	90.0	89.0	40.00%	40.06%	5.50%	5.49%	1.49%	0.11%
6	5	90.0	89.0	70.00%	69.87%	5.50%	5.49%	0.00%	0.08%
7	8	90.0	89.0	10.00%	9.79%	3.00%	2.99%	28.97%	0.08%
8	8	90.0	89.0	10.00%	10.28%	5.50%	5.55%	67.84%	0.10%
9	8	90.0	89.0	10.00%	10.16%	5.50%	5.49%	53.13%	0.08%
10	8	90.0	89.0	10.00%	10.00%	8.00%	8.03%	86.53%	0.08%
11	8	90.0	89.0	15.00%	15.03%	5.50%	5.49%	22.05%	0.08%
12	8	90.0	89.0	15.00%	14.98%	8.00%	8.03%	35.60%	0.08%
13	8	90.0	89.0	25.00%	24.99%	5.50%	5.51%	8.94%	0.09%
14	8	90.0	89.0	40.00%	40.01%	3.00%	2.99%	4.36%	0.15%
15	8	90.0	89.0	40.00%	41.65%	5.50%	5.77%	3.42%	0.11%
16	8	90.0	89.0	40.00%	40.13%	5.50%	5.49%	4.25%	0.11%
17	8	90.0	89.0	40.00%	40.11%	8.00%	8.00%	8.81%	0.11%
18	8	90.0	89.0	55.00%	54.96%	5.50%	5.51%	4.86%	0.14%
19	8	90.0	89.0	70.00%	69.97%	3.00%	2.99%	0.00%	0.31%
20	8	90.0	89.0	70.00%	71.68%	5.50%	5.94%	2.45%	0.13%
21	8	90.0	89.0	70.00%	69.95%	5.50%	5.49%	0.00%	0.13%
22	8	90.0	89.0	70.00%	69.95%	8.00%	8.00%	8.55%	0.21%
23	2	100.0	99.3	10.00%	10.42%	5.50%	5.75%	55.66%	0.21%
24	2	100.0	99.3	40.00%	40.20%	5.50%	5.50%	4.18%	0.11%
25	2	100.0	99.3	70.00%	70.15%	5.50%	5.50%	3.00%	0.11%
26	5	100.0	99.3	10.00%	10.02%	5.50%	5.50%	59.96%	0.21%
27	5	100.0	99.3	40.00%	40.20%	5.50%	5.50%	4.72%	0.11%
28	5	100.0	99.3	70.00%	70.05%	5.50%	5.50%	3.13%	0.08%
29	8	100.0	99.3	10.00%	9.98%	3.00%	3.01%	54.50%	0.29%
30	8	100.0	99.3	10.00%	10.44%	5.50%	5.79%	71.82%	0.15%
31	8	100.0	99.3	10.00%	9.99%	5.50%	5.50%	77.12%	0.11%
32	8	100.0	99.3	10.00%	9.98%	8.00%	7.99%	80.12%	0.13%
33	8	100.0	99.3	15.00%	15.01%	8.00%	7.99%	52.21%	0.08%
34	8	100.0	99.3	25.00%	25.08%	5.50%	5.48%	0.00%	0.08%
35	8	100.0	99.3	40.00%	40.05%	3.00%	3.01%	3.99%	0.08%
36	8	100.0	99.3	40.00%	40.59%	5.50%	5.70%	5.94%	0.08%
37	8	100.0	99.3	40.00%	40.25%	5.50%	5.50%	7.93%	0.08%
38	8	100.0	99.3	40.00%	40.08%	8.00%	8.01%	6.35%	0.13%
39	8	100.0	99.3	55.00%	55.01%	5.50%	5.48%	0.00%	0.11%
40	8	100.0	99.3	70.00%	70.02%	3.00%	3.01%	4.87%	0.11%
41	8	100.0	99.3	70.00%	70.41%	5.50%	5.75%	4.15%	0.13%
42	8	100.0	99.3	70.00%	70.53%	5.50%	5.50%	0.00%	0.21%
43	8	100.0	99.3	70.00%	69.99%	8.00%	8.01%	7.58%	0.21%
44	2	110.0	107.0	10.00%	10.06%	5.50%	5.50%	65.74%	0.21%
45	2	110.0	107.0	40.00%	40.06%	5.50%	5.50%	4.25%	0.11%
46	2	110.0	107.0	70.00%	69.96%	5.50%	5.50%	1.98%	0.08%
47	5	110.0	107.0	10.00%	9.96%	5.50%	5.50%	80.32%	0.21%
48	5	110.0	107.0	40.00%	40.07%	5.50%	5.50%	9.50%	0.11%
49	5	110.0	107.0	70.00%	70.12%	5.50%	5.50%	2.72%	0.08%
50	8	110.0	107.0	10.00%	10.33%	3.00%	3.00%	79.60%	0.29%
51	8	110.0	107.0	10.00%	10.10%	5.50%	5.58%	93.55%	0.14%
52	8	110.0	107.0	10.00%	10.01%	5.50%	5.50%	86.15%	0.11%
53	8	110.0	107.0	10.00%	10.02%	5.50%	5.49%	83.63%	0.21%
54	8	110.0	107.0	10.00%	10.72%	8.00%	8.00%	89.71%	0.11%
55	8	110.0	107.0	15.00%	15.04%	5.50%	5.49%	64.39%	0.08%
56	8	110.0	107.0	15.00%	15.07%	8.00%	8.00%	73.23%	0.08%
57	8	110.0	107.0	25.00%	25.08%	5.50%	5.49%	25.18%	0.13%
58	8	110.0	107.0	32.50%	32.56%	5.50%	5.50%	15.51%	0.10%
59	8	110.0	107.0	40.00%	40.01%	3.00%	3.00%	7.86%	0.08%
60	8	110.0	107.0	40.00%	39.59%	5.50%	5.42%	10.27%	0.08%
61	8	110.0	107.0	40.00%	40.07%	5.50%	5.50%	14.69%	0.09%
62	8	110.0	107.0	40.00%	40.07%	5.50%	5.49%	10.11%	0.09%
63	8	110.0	107.0	40.00%	39.87%	8.00%	8.00%	9.33%	0.11%
64	8	110.0	107.0	55.00%	55.04%	5.50%	5.49%	8.92%	0.11%
65	8	110.0	107.0	70.00%	69.96%	3.00%	3.00%	8.62%	0.11%
66	8	110.0	107.0	70.00%	70.32%	5.50%	5.63%	7.22%	0.14%
67	8	110.0	107.0	70.00%	69.99%	5.50%	5.50%	11.42%	0.21%
68	8	110.0	107.0	70.00%	70.00%	5.50%	5.49%	8.12%	0.21%
69	8	110.0	107.0	70.00%	70.08%	8.00%	8.00%	9.45%	0.21%

Table C2. Experimental results for the medium-sized system.

N°	Time, h	Temperature theo., °C	Temperature exp., °C	GTR/(GTR+Styrene) theo., %wt	GTR/(GTR+Styrene) exp., %wt	BPO/Styrene theo., %wt	BPO/Styrene exp., %wt	Conversion, %wt	Balance-Related Uncertainty, %wt
1	90	94.6	>98.16	10.00%	10.20%	3.00%	3.04%	18.43%	0.12%
2	90	93.4	>98.45	10.00%	10.15%	3.00%	3.01%	14.40%	0.12%
3	90	93.8	>109.47	10.00%	10.19%	5.50%	5.58%	49.20%	0.12%
4	90	na	na	10.00%	10.66%	5.50%	5.89%	37.88%	0.14%
5	90	92.2	>93	10.00%	10.15%	5.50%	5.57%	49.93%	0.13%
6	90	92.9	>146.30	10.00%	10.19%	8.00%	8.08%	72.52%	0.12%
7	90	95.7	>136.95	10.00%	10.28%	8.00%	8.16%	72.53%	0.12%
8	90	91.9	>94	15.00%	15.24%	5.50%	5.54%	10.40%	0.13%
9	90	92.3	>109.77	15.00%	15.25%	8.00%	8.07%	24.20%	0.13%
10	90	96.7	>115.47	25.00%	25.77%	4.25%	4.37%	6.11%	0.21%
11	90	95	>97.03	25.00%	25.60%	5.00%	5.13%	5.75%	0.21%
12	90	93.9	>95.97	25.00%	25.38%	6.00%	6.09%	5.03%	0.21%
13	90	93.9	>94.72	25.00%	25.35%	6.75%	6.83%	6.23%	0.21%
14	90	na	na	30.00%	30.47%	5.50%	5.56%	2.75%	0.22%
15	90	90.5	>91	30.00%	30.37%	5.50%	5.51%	2.64%	0.22%
16	90	92.5	>93.30	40.00%	40.47%	5.50%	5.54%	3.54%	0.35%
17	90	90.7	>95.85	40.00%	40.73%	8.00%	8.16%	5.39%	0.35%
18	90	na	na	70.00%	72.46%	5.50%	5.88%	6.99%	1.19%
19	90	89.2	>92	70.00%	73.21%	5.50%	6.32%	5.87%	1.22%
20	100	103.6	>105.69	10.00%	10.18%	3.00%	3.04%	45.04%	0.12%
21	100	104.4	>123.65	10.00%	10.20%	5.50%	5.56%	69.92%	0.12%
22	100	103.9	>129.84	10.00%	10.62%	8.00%	8.45%	91.82%	0.13%
23	100	102.6	>143.30	10.00%	10.32%	8.00%	8.20%	78.12%	0.13%
24	100	102.2	>118.83	10.00%	10.16%	8.00%	8.08%	83.37%	0.13%
25	100	103	>104.38	15.00%	15.20%	5.50%	5.49%	19.42%	0.13%
26	100	na	na	15.00%	15.22%	5.50%	5.53%	18.57%	0.13%
27	100	101.7	>109.85	15.00%	15.23%	8.00%	8.08%	36.72%	0.13%
28	100	na	na	20.00%	21.86%	5.50%	6.11%	6.61%	0.14%
29	100	104	>105	20.00%	20.24%	5.50%	5.57%	9.81%	0.13%
30	100	102	>103.96	20.00%	20.28%	8.00%	8.07%	12.43%	0.14%
31	100	106	>106.85	25.00%	25.52%	4.25%	4.31%	16.18%	0.20%
32	100	103.7	>104.90	25.00%	25.12%	5.00%	4.99%	9.70%	0.21%
33	100	104.3	>105.29	25.00%	25.17%	5.00%	5.03%	12.52%	0.20%
34	100	100.2	>106.49	25.00%	25.34%	6.00%	6.05%	6.07%	0.18%
35	100	104.5	>105.65	25.00%	25.36%	6.00%	6.06%	8.68%	0.18%
36	100	106.7	>115.21	25.00%	26.05%	6.75%	7.06%	16.49%	0.21%
37	100	na	na	30.00%	30.05%	3.00%	3.01%	4.13%	0.20%
38	100	101.4	>103	30.00%	30.33%	3.00%	3.00%	3.85%	0.21%
39	100	na	na	30.00%	30.30%	5.50%	5.53%	5.02%	0.21%
40	100	101.1	>103	30.00%	30.40%	5.50%	5.56%	4.58%	0.21%
41	100	na	na	30.00%	30.24%	8.00%	8.05%	4.77%	0.21%
42	100	100.4	>103	30.00%	30.72%	8.00%	8.19%	6.15%	0.21%
43	100	103.3	>105.66	32.50%	33.32%	5.50%	5.63%	6.43%	0.27%
44	100	103.3	>104.14	32.50%	32.73%	6.25%	6.25%	7.49%	0.26%
45	100	101.9	>105.63	35.00%	36.84%	4.00%	4.30%	5.30%	0.28%
46	100	103.8	>105.61	35.00%	35.92%	7.00%	7.19%	7.74%	0.28%
47	100	101.6	>105.65	40.00%	40.43%	3.00%	3.03%	5.48%	0.35%
48	100	na	na	40.00%	40.75%	5.50%	5.63%	4.37%	0.34%
49	100	99.6	>101	40.00%	40.62%	5.50%	5.56%	4.20%	0.33%
50	100	104.5	>106.32	40.00%	41.69%	6.25%	6.55%	8.69%	0.36%
51	100	102	>102.70	40.00%	40.90%	8.00%	8.13%	7.79%	0.36%
52	100	na	na	70.00%	72.44%	5.50%	6.05%	8.07%	1.14%
53	100	101.2	>102	70.00%	72.69%	5.50%	6.00%	8.43%	1.15%
54	100	99.4	>103.70	70.00%	73.56%	8.00%	9.16%	12.16%	1.30%
55	110	114.8	>115.40	25.00%	25.25%	4.25%	4.18%	22.11%	0.20%
56	110	112.8	>113.52	25.00%	25.25%	5.00%	5.04%	18.17%	0.20%
57	110	112.9	>113.60	25.00%	25.17%	6.75%	6.78%	20.68%	0.20%
58	110	112.9	>113.74	32.50%	32.94%	5.50%	5.52%	10.84%	0.26%
59	110	113.3	>114.45	32.50%	33.11%	6.25%	6.29%	11.44%	0.26%
60	110	113.7	>114.96	35.00%	35.91%	7.00%	7.16%	13.46%	0.33%

Appendix D. Detailed ML Results

Table D1. Detailed metrics (R^2, RMSE and MAE) obtained for the ten screened models with default HPs.

ML Algorithm	R^2 Train	Test	RMSE Train	Test	MAE Train	Test
LR	0.697 ± 0.016	0.619 ± 0.107	0.168 ± 0.003	0.177 ± 0.013	0.148 ± 0.003	0.156 ± 0.011
Ridge	0.697 ± 0.016	0.621 ± 0.101	0.168 ± 0.003	0.177 ± 0.013	0.148 ± 0.003	0.156 ± 0.012
Lasso	0.000 ± 0.000	–0.096 ± 0.069	0.306 ± 0.010	0.308 ± 0.038	0.270 ± 0.011	0.273 ± 0.018
SVR	0.905 ± 0.013	0.762 ± 0.066	0.094 ± 0.009	0.142 ± 0.029	0.078 ± 0.007	0.111 ± 0.021
GP	1.000 ± 0.000	–1736 ± 1572	0.000 ± 0.000	8.644 ± 6.903	0.000 ± 0.000	3.592 ± 2.829
kNN	0.769 ± 0.043	0.554 ± 0.206	0.146 ± 0.012	0.193 ± 0.064	0.104 ± 0.011	0.137 ± 0.047
DT	1.000 ± 0.000	0.843 ± 0.050	0.000 ± 0.000	0.116 ± 0.027	0.000 ± 0.000	0.075 ± 0.017
RF	0.986 ± 0.002	0.899 ± 0.042	0.036 ± 0.003	0.094 ± 0.031	0.023 ± 0.002	0.063 ± 0.022
GB	1.000 ± 0.000	0.927 ± 0.032	0.007 ± 0.000	0.079 ± 0.026	0.005 ± 0.000	0.054 ± 0.016
MLP	0.925 ± 0.012	0.804 ± 0.071	0.083 ± 0.008	0.127 ± 0.030	0.061 ± 0.006	0.096 ± 0.025

Table D2. Detailed metrics (R^2, RMSE and MAE) obtained for SVR, RF, GB and MLP with optimized HPs.

	SVR	RF	GB	MLP
R^2 train	0.959 ± 0.019	0.985 ± 0.002	0.996 ± 0.002	0.992 ± 0.008
R^2 test	0.806 ± 0.119	0.878 ± 0.049	0.905 ± 0.043	0.898 ± 0.045
RMSE train	0.061 ± 0.015	0.038 ± 0.003	0.018 ± 0.005	0.025 ± 0.012
RMSE test	0.124 ± 0.048	0.103 ± 0.033	0.088 ± 0.025	0.090 ± 0.017
MAE train	0.031 ± 0.007	0.025 ± 0.002	0.014 ± 0.003	0.016 ± 0.009
MAE test	0.090 ± 0.035	0.070 ± 0.023	0.059 ± 0.014	0.062 ± 0.016

Appendix E. Ways of Improvement for the ML Model

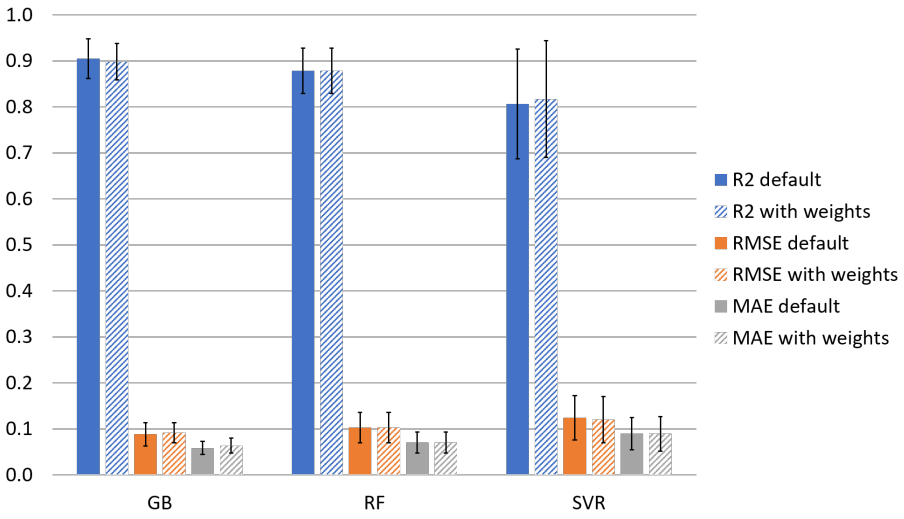


Figure E1. Comparison of test performances of GB, RF and SVR with optimized HPs, without weighted vs. with weighted samples.

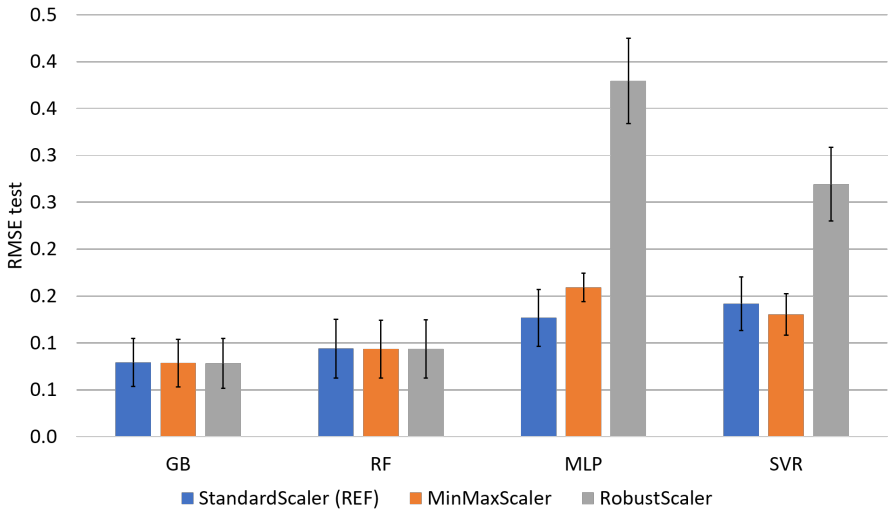


Figure E2. Scaler impact on test RMSE, without HP optimization.

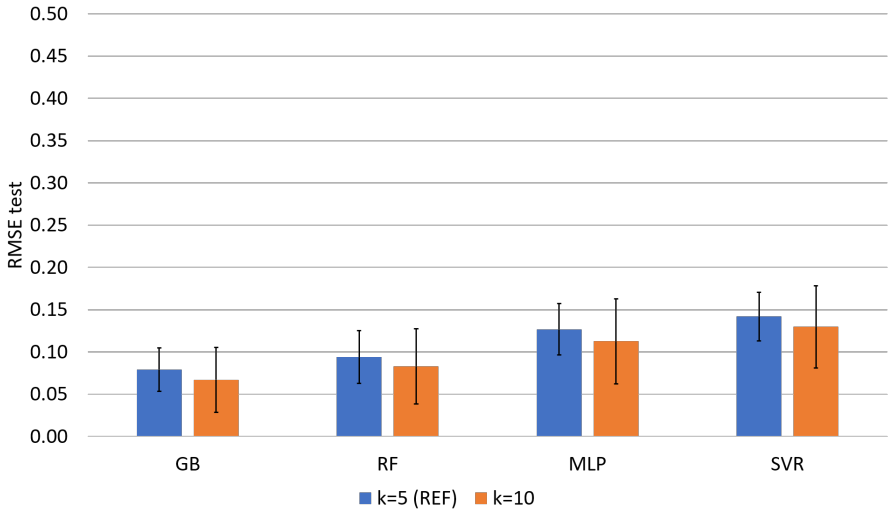


Figure E3. Partitioning train/test impact on test RMSE, without HP optimization.

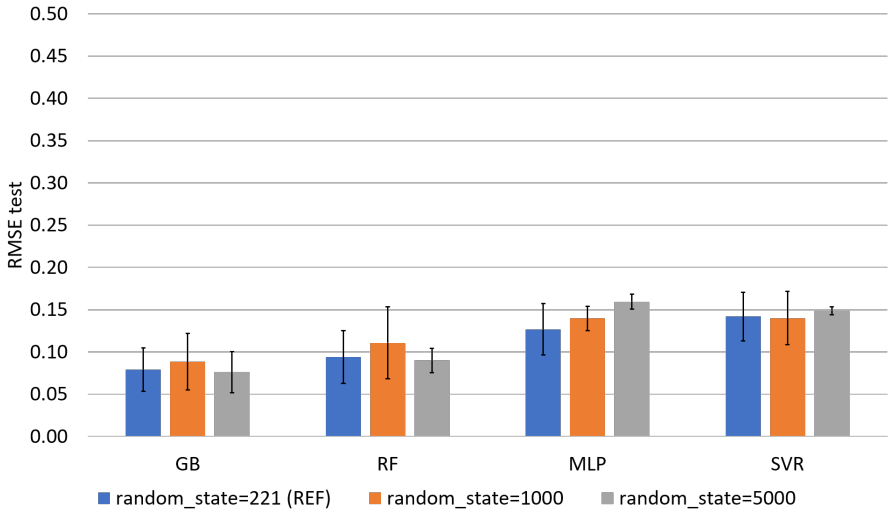


Figure E4. Random state impact on test RMSE, without HP optimization.

References

1. Abbas-Abadi, M.S.; Kusenberg, M.; Shirazi, H.M.; Goshayeshi, B.; Van Geem, K.M. Towards full recyclability of end-of-life tires: Challenges and opportunities. *J. Clean. Prod.* **2022**, *374*, 134036. [CrossRef]
2. Hejna, A.; Korol, J.; Przybysz-Romatowska, M.; Zedler, Ł.; Chmielnicki, B.; Formela, K. Waste tire rubber as low-cost and environmentally-friendly modifier in thermoset polymers—A review. *Waste Manag.* **2020**, *108*, 106–118. [CrossRef] [PubMed]
3. Fazli, A.; Rodrigue, D. Recycling waste tires into ground tire rubber (Gtr)/rubber compounds: A review. *J. Compos. Sci.* **2020**, *4*, 103. [CrossRef]
4. Ramarad, S.; Khalid, M.; Ratnam, C.T.; Chuah, A.L.; Rashmi, W. Waste tire rubber in polymer blends: A review on the evolution, properties and future. *Prog. Mater. Sci.* **2015**, *72*, 100–140. [CrossRef]
5. Araujo-Morera, J.; Verdugo-Manzanares, R.; González, S.; Verdejo, R.; Lopez-Manchado, M.A.; Santana, M.H. On the use of mechano-chemically modified ground tire rubber (Gtr) as recycled and sustainable filler in styrene-butadiene rubber (sbr) composites. *J. Compos. Sci.* **2021**, *5*, 68. [CrossRef]
6. Phiri, M.M.; Phiri, M.J.; Formela, K.; Wang, S.; Hlangothi, S.P. Grafting and reactive extrusion technologies for compatibilization of ground tyre rubber composites: Compounding, properties, and applications. *J. Clean. Prod.* **2022**, *369*, 133084. [CrossRef]
7. Colom, X.; Cañavate, J.; Carrillo-Navarrete, F. Towards Circular Economy by the Valorization of Different Waste Subproducts through Their Incorporation in Composite Materials: Ground Tire Rubber and Chicken Feathers. *Polymers* **2022**, *14*, 1090. [CrossRef]
8. He, M.; Gu, K.; Wang, Y.; Li, Z.; Shen, Z.; Liu, S.; Wei, J. Development of high-performance thermoplastic composites based on polyurethane and ground tire rubber by in-situ synthesis. *Resour. Conserv. Recycl.* **2021**, *173*, 105713. [CrossRef]
9. Archibong, F.N.; Sanusi, O.M.; Médéric, P.; Hocine, N.A. An overview on the recycling of waste ground tyre rubbers in thermoplastic matrices: Effect of added fillers. *Resour. Conserv. Recycl.* **2021**, *175*, 105894. [CrossRef]
10. Liu, S.; Peng, Z.; Zhang, Y.; Rodrigue, D.; Wang, S. Compatibilized thermoplastic elastomers based on highly filled polyethylene with ground tire rubber. *J. Appl. Polym. Sci.* **2022**, *139*, e52999. [CrossRef]
11. Fan, P.; Lu, C. Surface Graft Copolymerization of Poly(methyl methacrylate) onto Waste Tire Rubber Powder through Ozonization. *J. Appl. Polym. Sci.* **2011**, *122*, 2262–2270. [CrossRef]
12. Coiai, S.; Passaglia, E.; Ciardelli, F.; Tirelli, D.; Peruzzotti, F.; Resmini, E. Modification of cross-linked rubber particles by free radical polymerization. *Macromol. Symp.* **2006**, *234*, 193–202. [CrossRef]
13. Sulcis, R.; Lotti, L.; Coiai, S.; Ciardelli, F.; Passaglia, E. Novel HDPE/ground tyre rubber composite materials obtained through in-situ polymerization and polymerization filling technique. *J. Appl. Polym. Sci.* **2014**, *131*, 1–13. [CrossRef]
14. Sulcis, R.; Vizza, F.; Oberhauser, W.; Ciardelli, F.; Spiniello, R.; Dintcheva, N.T.; Passaglia, E. Recycling ground tire rubber (GTR) scraps as high-impact filler of in situ produced polyketone matrix. *Polym. Adv. Technol.* **2014**, *25*, 1060–1068. [CrossRef]
15. Tsagkalias, I.S.; Vlachou, A.; Verros, G.D.; Achilias, D.S. Effect of graphene oxide or functionalized graphene oxide on the copolymerization kinetics of Styrene/n-butyl methacrylate. *Polymers* **2019**, *11*, 999. [CrossRef] [PubMed]
16. Potts, J.R.; Lee, S.H.; Alam, T.M.; An, J.; Stoller, M.D.; Piner, R.D.; Ruoff, R.S. Thermomechanical properties of chemically modified graphene/poly(methyl methacrylate) composites made by in situ polymerization. *Carbon* **2011**, *49*, 2615–2623. [CrossRef]
17. Tripathi, S.N.; Saini, P.; Gupta, D.; Choudhary, V. Electrical and mechanical properties of PMMA/reduced graphene oxide nanocomposites prepared via in situ polymerization. *J. Mater. Sci.* **2013**, *48*, 6223–6232. [CrossRef]
18. Wang, J.; Hu, H.; Wang, X.; Xu, C.; Zhang, M.; Shang, X. Preparation and Mechanical and Electrical Properties of Graphene Nanosheets–Poly(methyl methacrylate) Nanocomposites via In Situ Suspension Polymerization Jingchao. *J. Appl. Polym. Sci.* **2011**, *122*, 1866–1871. [CrossRef]
19. Feng, L.; Guan, G.; Li, C.; Zhang, D.; Xiao, Y.; Zheng, L.; Zhu, W. In situ synthesis of poly(methyl methacrylate)/graphene oxide nanocomposites using thermal-initiated and graphene oxide-initiated polymerization. *J. Macromol. Sci. Part A Pure Appl. Chem.* **2013**, *50*, 720–727. [CrossRef]
20. Michailidis, M.; Verros, G.D.; Deliyanni, E.A.; Andriotis, E.G.; Achilias, D.S. An experimental and theoretical study of butyl methacrylate in situ radical polymerization kinetics in the presence of graphene oxide nanoadditive. *J. Polym. Sci. Part A Polym. Chem.* **2017**, *55*, 1433–1441. [CrossRef]
21. Tsagkalias, I.S.; Papadopoulou, S.; Verros, G.D.; Achilias, D.S. Polymerization Kinetics of n-Butyl Methacrylate in the Presence of Graphene Oxide Prepared by Two Different Oxidation Methods with or without Functionalization. *Ind. Eng. Chem. Res.* **2018**, *57*, 2449–2460. [CrossRef]
22. Funck, A.; Kaminsky, W. Polypropylene carbon nanotube composites by in situ polymerization. *Compos. Sci. Technol.* **2007**, *67*, 906–915. [CrossRef]
23. Jiang, X.; Bin, Y.; Matsuo, M. Electrical and mechanical properties of polyimide-carbon nanotubes composites fabricated by in situ polymerization. *Polymer* **2005**, *46*, 7418–7424. [CrossRef]
24. Verros, G.D.; Achilias, D.S. Toward the development of a mathematical model for the bulk in situ radical polymerization of methyl methacrylate in the presence of nano-additives. *Can. J. Chem. Eng.* **2016**, *94*, 1783–1791. [CrossRef]
25. Yeh, J.M.; Liou, S.J.; Lai, M.C.; Chang, Y.W.; Huang, C.Y.; Chen, C.P.; Jaw, J.H.; Tsai, T.Y.; Yu, Y.H. Comparative studies of the properties of poly(methyl methacrylate)-clay nanocomposite materials prepared by in situ emulsion polymerization and solution dispersion. *J. Appl. Polym. Sci.* **2004**, *94*, 1936–1946. [CrossRef]

26. Meng, X.; Wu, H.; Storti, G.; Morbidelli, M. Effect of Dispersed Polymeric Nanoparticles on the Bulk Polymerization of Methyl Methacrylate. *Macromolecules* **2016**, *49*, 7758–7766. [CrossRef]
27. Florez, D.; Hoppe, S.; Hu, G.H.; Meimaroglou, D. Radical bulk polymerization of styrene in the presence of rubber particles from recycled tires: A kinetic study using DSC. *J. Therm. Anal. Calorim.* **2020**, *143*, 3073–3084. [CrossRef]
28. Florez Parra, D.C. Effects of the Presence of Recycled Tire Powders on the Kinetics of the Radical Polymerization of Styrene and the Properties of the Resulting Materials. Ph.D. Thesis, Université de Lorraine, Lorraine, France, 2019; pp. 1–197.
29. Meimaroglou, D.; Florez, D.; Hu, G.H. A kinetic modeling framework for the peroxide-initiated radical polymerization of styrene in the presence of rubber particles from recycled tires. *Chem. Eng. Sci.* **2022**, *248*, 117137. [CrossRef]
30. Cameron, G.G.; Qureshi, M.Y. Free Radical Grafting of Monomers to Polydienes–4. Kinetics and Mechanism of Methyl Methacrylate Grafting to Polybutadiene. *J. Polym. Sci. Part A-1 Polym. Chem.* **1980**, *18*, 3149–3161. [CrossRef]
31. Estenoz, D.A.; Meira, G.R.; Gomez, N.; Oliva, H.M. Mathematical model of a continuous industrial high-impact polystyrene process. *AIChE J.* **1998**, *44*, 427–441. [CrossRef]
32. Meira, G.R.; Luciani, C.V.; Estenoz, D.A. Continuous Bulk Process for the Production of High-Impact Polystyrene: Recent Developments in Modeling and Control. *Macromol. React. Eng.* **2007**, *1*, 25–39. [CrossRef]
33. Zhu, C.X.; Wu, Y.Y.; Figueira, F.L.; Van Steenberge, P.H.; D'hooge, D.R.; Zhou, Y.N.; Luo, Z.H. Sensitivity analysis of isothermal free radical induced grafting through application of the distribution—Numerical fractionation—Method of moments. *Chem. Eng. J.* **2022**, *444*, 136595. [CrossRef]
34. Xu, P.; Chen, H.; Li, M.; Lu, W. New Opportunity: Machine Learning for Polymer Materials Design and Discovery. *Adv. Theory Simul.* **2022**, *5*, 2100565. [CrossRef]
35. Trinh, C.; Meimaroglou, D.; Hoppe, S. Machine learning in chemical product engineering: The state of the art and a guide for newcomers. *Processes* **2021**, *9*, 1456. [CrossRef]
36. Zhang, J.L.; Chen, H.X.; Ke, C.M.; Zhou, Y.; Lu, H.Z.; Wang, D.L. Graft polymerization of styrene onto waste rubber powder and surface characterization of graft copolymer. *Polym. Bull.* **2012**, *68*, 789–801. [CrossRef]
37. Fazli, A.; Rodrigue, D. Effect of ground tire rubber (Gtr) particle size and content on the morphological and mechanical properties of recycled high-density polyethylene (rhdpe)/gtr blends. *Recycling* **2021**, *6*, 44. [CrossRef]
38. Zhang, J.; Chen, H.; Zhou, Y.; Ke, C.; Lu, H. Compatibility of waste rubber powder/polystyrene blends by the addition of styrene grafted styrene butadiene rubber copolymer: Effect on morphology and properties. *Polym. Bull.* **2013**, *70*, 2829–2841. [CrossRef]
39. Aggarwal, P.K.; Karmarkar, A.; Taqui, S.N. Grafting of Styrene Onto Cellulose. *Int. Res. J. Eng. Technol.* **2018**, *2*, 1081–1089.
40. Hejna, A.; Klein, M.; Saeb, M.R.; Formela, K. Towards understanding the role of peroxide initiators on compatibilization efficiency of thermoplastic elastomers highly filled with reclaimed GTR. *Polym. Test.* **2019**, *73*, 143–151. [CrossRef]
41. Liu, H.L.; Wang, X.P.; Jia, D.M. Recycling of waste rubber powder by mechano-chemical modification. *J. Clean. Prod.* **2020**, *245*, 118716. [CrossRef]
42. Fletes, R.C.; López, E.O.; Gudiño, P.O.; Mendizábal, E.; Núñez, R.G.; Rodrigue, D. Ground tire rubber/polyamide 6 thermoplastic elastomers produced by dry blending and compression molding. *Prog. Rubber Plast. Recycl. Technol.* **2022**, *38*, 38–55. [CrossRef]
43. Ku, H. Notes on the use of propagation of error formulas. *J. Res. Natl. Bur. Stand. Sect. C Eng. Instrum.* **1966**, *70C*, 263. [CrossRef]
44. Rodriguez-Galiano, V.; Sanchez-Castillo, M.; Chica-Olmo, M.; Chica-Rivas, M. Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geol. Rev.* **2015**, *71*, 804–818. [CrossRef]
45. Freund, Y.; Schapire, R.E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
46. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
47. Friedman, J.H. Stochastic gradient boosting. *Comput. Stat. Data Anal.* **2002**, *38*, 367–378. [CrossRef]
48. Natekin, A.; Knoll, A. Gradient boosting machines, a tutorial. *Front. Neurorobotics* **2013**, *7*, 21. [CrossRef]
49. Bentéjac, C.; Csörgő, A.; Martínez-Muñoz, G. A comparative analysis of gradient boosting algorithms. *Artif. Intell. Rev.* **2021**, *54*, 1937–1967. [CrossRef]
50. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
51. Liaw, A.; Wiener, M. Classification and Regression by randomForest. *R News* **2002**, *2*, 18–22.
52. Zhang, C.; Ma, Y. *Ensemble Machine Learning*; Springer: New York, NY, USA, 2012. [CrossRef]
53. Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J.C.; Sheridan, R.P.; Feuston, B.P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1947–1958. [CrossRef] [PubMed]
54. Segal, M.R. *Machine Learning Benchmarks and Random Forest Regression Publication Date Machine Learning Benchmarks and Random Forest Regression*; Center for Bioinformatics and Molecular Biostatistics: San Francisco, CA, USA, 2004; p. 15.
55. Krogh, A. What are artificial neural networks? *Nat. Biotechnol.* **2008**, *26*, 195–197. [CrossRef]
56. Murtagh, F. Multilayer perceptrons for classification and regression. *Neurocomputing* **1991**, *2*, 183–197. [CrossRef]
57. Bishop, C.M. Neural networks and their applications. *Rev. Sci. Instrum.* **1998**, *65*, 1803–1832. [CrossRef]
58. Gasteiger, J.; Zupan, J. Neural Networks in Chemistry. *Angew. Chem. Int. Ed. Engl.* **1993**, *32*, 503–527. [CrossRef]
59. Gardner, M.W.; Dorling, S.R. Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences. *Atmos. Environ.* **1998**, *32*, 2627–2636. [CrossRef]

60. Zhang, Z.; Friedrich, K. Artificial neural networks applied to polymer composites: A review. *Compos. Sci. Technol.* **2003**, *63*, 2029–2044. [CrossRef]
61. Paliwal, M.; Kumar, U.A. Neural networks and statistical techniques: A review of applications. *Expert Syst. Appl.* **2009**, *36*, 2–17. [CrossRef]
62. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.E.; Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* **2018**, *4*, e00938. [CrossRef] [PubMed]
63. Awad, M.; Khanna, R. *Efficient Learning Machines*; Apress: Berkeley, CA, USA, 2015.
64. Ge, Z.; Song, Z.; Ding, S.X.; Huang, B. Data Mining and Analytics in the Process Industry: The Role of Machine Learning. *IEEE Access* **2017**, *5*, 20590–20616. [CrossRef]
65. Zendehboudi, S.; Rezaei, N.; Lohi, A. Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review. *Appl. Energy* **2018**, *228*, 2539–2566. [CrossRef]
66. Golkarnarenji, G.; Naebe, M.; Badii, K.; Milani, A.S.; Jazar, R.N.; Khayyam, H. A machine learning case study with limited data for prediction of carbon fiber mechanical properties. *Comput. Ind.* **2019**, *105*, 123–132. [CrossRef]
67. Vapnik, V.N. *The Nature of Statistical Learning Theory*; Springer: New York, NY, USA, 1995.
68. Drucker, H.; Surges, C.J.; Kaufman, L.; Smola, A.; Vapnik, V. Support vector regression machines. *Adv. Neural Inf. Process. Syst.* **1997**, *1*, 155–161.
69. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
70. Welling, M. Support Vector Regression. Available online: https://www.academia.edu/326409 (accessed on 1 December 2022).
71. Yang, L.; Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, *415*, 295–316. [CrossRef]
72. Vabalas, A.; Gowen, E.; Poliakoff, E.; Casson, A.J. Machine learning algorithm validation with a limited sample size. *PLoS ONE* **2019**, *14*, e0224365. [CrossRef]
73. Cawley, G.C.; Talbot, N.L. On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.* **2010**, *11*, 2079–2107.
74. Zhang, Y.; Ling, C. A strategy to apply machine learning to small datasets in materials science. *npj Comput. Mater.* **2018**, *4*, 28–33. [CrossRef]
75. Shahvandi, M.K.; Soja, B. Inclusion of data uncertainty in machine learning and its application in geodetic data science, with case studies for the prediction of Earth orientation parameters and GNSS station coordinate time series. *Adv. Space Res.* **2022**, *70*, 563–575. [CrossRef]
76. Czarnecki, W.M.; Podolak, I.T. Machine Learning with Known Input Data Uncertainty Measure. In Proceedings of the 12th IFIP TC 8 International Conference, CISIM 2013, Krakow, Poland, 25–27 September 2013; Volume 8104, pp. 379–388. [CrossRef]
77. Alizadehsani, R.; Roshanzamir, M.; Hussain, S.; Khosravi, A.; Koohestani, A.; Zangooei, M.H.; Abdar, M.; Beykikhoshk, A.; Shoeibi, A.; Zare, A.; et al. Handling of uncertainty in medical data using machine learning and probability theory techniques: A review of 30 years (1991–2020). *Ann. Oper. Res.* **2021**. [CrossRef] [PubMed]
78. Psaros, A.F.; Meng, X.; Zou, Z.; Guo, L.; Karniadakis, G.E. Uncertainty Quantification in Scientific Machine Learning: Methods, Metrics, and Comparisons. *J. Comput. Phys.* **2023** , *477*, 111902. [CrossRef]

Chapter 5

Hybrid modeling and Gaussian Processes in polymer engineering

Contents

5.1 Introduction

In view of the complex problems encountered in chemical process and product engineering, it becomes quickly appealing to resort to machine learning (ML) data-driven models. However, not only the latter should not be considered systematically (cf. Chapter 1), but also it would be an error to ignore the knowledge accumulated over many years of scientific research, for example on the understanding of the physico-chemical mechanisms. In this sense, hybrid modeling (i.e., combining ML and knowledge-based models) is of great interest in chemical process and product engineering, where prior knowledge is often available and data are of limited quantity. In particular, the developed models benefit from a better interpretability and more physically-consistent predictions (thanks to the incorporation of knowledge) which can facilitate their acceptance in more critical applications in comparison with ML black-box models. At the same time, knowledge-based models that are developed for a given process under specific operating conditions can show deviations when being applied to the same process but under modified conditions. This is mainly due to the unavoidable uncertainties that accompany the values of the model parameters. These uncertainties reflect both the experimental error, inherent to the measurements that have been used during the parametric estimation process, as well as the eventual structural inefficiencies of the mathematical model. In other words, if the measurements used for the development of a model contain deviations due to impurities or due to an overlooked factor during the experimental process, these will be incorporated in the predictions of the model. In the same sense, if a phenomenon is negligible under the conditions of the used measurements, the model might fail to incorporate it at all. Accordingly, when the same model is confronted with new measurements that follow a different protocol (e.g., systematic evaluation of impurities, better control of all the affecting factors, etc.) or have been carried out under a different range of conditions (i.e., leading to the apparition of new mechanisms), it is bound to decrease its accuracy or even fail in its predictions. In these cases, the standard strategy for knowledge-based models is a new parametric estimation in the light of the new data. This is where ML methods can help to account for this deviation, thus avoiding the time-consuming procedure of re-developing a knowledge-based model anytime the process conditions are modified, as is often the case in the small batch production of specialty products or other products with changing end-of-use properties.

In the previous chapter, "pure" ML models were developed for the prediction of styrene conversion based on operating conditions, in the styrene-ground tire rubber (GTR) radical graft polymerization reaction. Indeed, the available kinetic models were not considered due to the unknown composition and structure of GTR which complexifies the development of such models. The goal of the present chapter is therefore to investigate the combination of ML models with knowledge-based models as part of a hybrid modeling framework. Concerning the ML part, Gaussian processes (GP) were employed exclusively for their interesting ability in providing predictions accompanied by their uncertainties and in incorporating prior information. As for the knowledge-based part, the kinetic model developed during a precedent thesis was exploited. To build the hybrid model, data was collected from different literature works as well as from a new experimental procedure. Besides, the obtained performance was compared to those of the pure ML and the knowledge-based models.

This chapter essentially constitutes a preliminary work, to be further pursued for later publication. Additionally, this work focuses on the prediction of styrene conversion, but could be extended to other properties such as the grafting efficiency of polystyrene (PS) on GTR or the

average molecular weight of PS.

5.2 Literature review

5.2.1 Hybrid modeling

In this work, hybrid modeling (a.k.a. grey-box, semi-empirical, semi-mechanistic modeling) refers to the association of knowledge-based modeling (white-box, WB) with data-driven modeling (black-box, BB).[1] Several reported works provide a good overview of these concepts and terminology and display the numerous applications of hybrid modeling in the industry, for chemical, petroleum and energy processes, and in smart manufacturing [76, 90, 93, 95]. These models have gained increasing interest over the last two decades, especially with the increasing willingness to build more interpretable ML models (cf. OECD's principles for QSAR validation in Chapters 2 and 3, and recent studies [23, 25, 59, 61]). Concretely, the goal of hybrid models is to exploit all the information available about a given system, namely the *a priori* knowledge and the knowledge that is hidden within the data, which generally results in improved models in comparison to models that are exclusively BB or WB. In particular, hybrid models provide a good compromise between BB and WB models in terms of interpretability, extrapolation capability, computation time, model complexity and data requirement, as summarized in Table 5.1.

Table 5.1: Comparison of white-box (WB), black-box (BB) and hybrid models.

Models	Physical interpretability	Extrapolation capability	Computation time	Model complexity	Data requirement
WB	Good	Good	High	High	Low
BB	None	Poor	Low	Low	High
Hybrid	Medium	Medium	Medium	Medium	Medium

On the one hand, WB models (a.k.a. mechanistic, first-principles, phenomenological, physics-based models) rely on the description of the system's behavior by means of physical/chemical concepts. Examples in chemical engineering problems include material and energy balances, thermodynamics as well as transport and kinetics laws. For more complex systems (e.g., composed of multiple ingredients and/or structures), the development of WB models becomes however a tedious task with the need to state idealized assumptions and the risk to omit interactions between parts of the system. This leads to deviations that could eventually be taken into account by a BB model, as part of a hybrid modeling framework. These deviations are quite frequent in the case of batch processes where the operating conditions and product requirements may vary, hence the interest in hybrid models in such studies. As they rely on rigorous knowledge, WB models benefit from good interpretabilty and extrapolation capability and do not require important amounts of data. The computation time and model complexity vary depending on the problem, but in general, the computation time is often rather high (e.g., when the resolution

[1] Note that the term "white-box", although prevailing in the literature, does not capture the essence of the nature of these models that allow a clear "view" and understanding of the mechanisms taking place "inside the box" (the process). As such, the term "glass-box" would be more appropriate as it contains the characteristic of transparency. Nonetheless, the term WB being widely employed in the literature, also to emphasize the contrast with BB models, it will be adopted in this work as well.

of large systems of differential or algebraic-differential equations must be solved numerically) and the model can become rather complex (e.g., when different length- and time-scales of the system are considered).

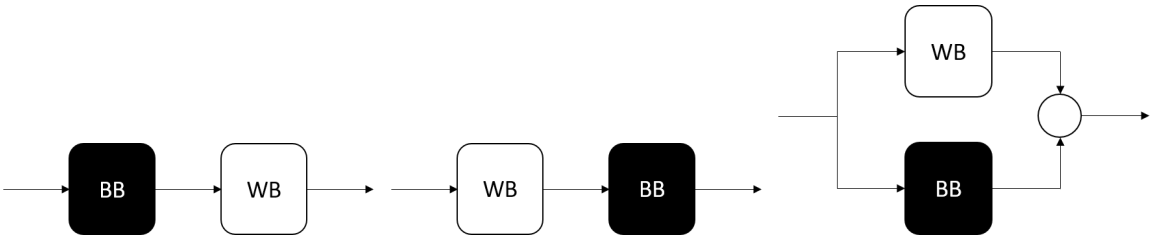
On the other hand, BB models (a.k.a statistical, empirical) do not necessitate significant prior knowledge about the system's behavior. Instead, they rely essentially on the available or collected data, describing the behavior of the system in a specific range of experimental conditions. In this sense, these data need to meet certain conditions of quantity, quality and relevant information content (with respect to the studied system's behavior) to enable the construction of consistent and applicable models. BB models are generally simpler than WB models and, consequently, faster to compute. However, the data-driven character of these models limits their physical interpretation and extrapolation capability. As a result, using pure BB models can lead to physically unrealistic predictions and this is where the combination with a WB model becomes interesting.

In terms of structure, hybrid models can be classified into different categories, among which the most commonly encountered are the serial and the parallel configurations (see Figure 5.1). Several reviews on the structures of hybrid models were provided in the literature and the general concepts are summarized in the following [76, 90, 93, 95]. The serial approach combines the BB and WB models sequentially, either in the order BB/WB or WB/BB. The first configuration (BB/WB) is more popular and is used when the system's behavior can not be fully described by WB models (e.g., not enough understanding/knowledge, too complex behavior, significant uncertainties). In this sense, BB models are first used to estimate the missing part, which is then incorporated in WB models. An example is the case of material or energy balances (WB) in which the reaction kinetics are represented by BB models [7]. Concerning the second configuration (WB/BB), BB models receive information from both the collected data and the WB models. For example, WB models can be used to determine some inputs for BB models, to generate additional data (e.g., physics-based simulations) or to add constraints on the BB models. Another example of WB/BB configuration consists in using the BB model to correct the residual error between the WB model predictions and the observations. All these examples are further illustrated in Sections 5.2.2 and 5.2.3 with applications of Gaussian Processes-based hybrid models in various fields and of hybrid models in polymer engineering.

In the parallel approach, BB and WB models are used either in a cooperative or competitive manner [76]. The cooperative case is related to the WB/BB example described previously, the BB model tries to learn the residual between the WB model and the observations, thus describing the deviation resulting from the effects that were not considered in the WB model, wrong assumptions or uncertainties in the system. To obtain the final prediction (hybrid model prediction), the WB and BB predictions are combined, for example via an addition [93]. As for the competitive case, both WB and BB models are trained in parallel on the same data and the final prediction is obtained by combining both predictions via a weighing scheme [8].

The choice of the serial or the parallel configuration will depend on the content of the WB model [76, 90, 95]. If the latter is incomplete/inaccurate in terms of the description of the system, the parallel cooperative configuration can help in correcting the eventual deviation between the WB model predictions and the observations, while the competitive configuration will require a weighting scheme giving less weight to the WB model. However, if the WB model has a good prediction accuracy, the serial configuration BB/WB is more likely to provide better results and extrapolation capabilities than the parallel one. As for the serial WB/BB configuration (e.g,

inputs of the BB model determined by a BB model, BB model trained on data generated by a WB model), an accurate WB model is necessary to obtain an accurate and reliable BB model. Hybrid models generally outperform WB models, under the condition that the latter provide a good representation of the system's behavior. If not, hybrid models can be outperformed by BB models. More generally, there is a lack of studies that compare the different types of hybrid structures for different levels of knowledge contained in the WB model. In this preliminary work, a simple hybrid configuration (cf. Section 5.3.2.3) was chosen but further comparison with other hybrid structures and with different levels of knowledge contained in the WB model are envisioned as future steps.



(a) Serial BB/WB (b) Serial WB/BB (c) Parallel

Figure 5.1: Hybrid model structures.

5.2.2 Gaussian Processes in hybrid modeling

Gaussian Processes (GP) are a specific type of ML method, widely used in several scientific areas for regression tasks in particular due to its ability in providing not only predictions but also their uncertainties, which better represents physical reality. Indeed, while traditional ML methods focus in predicting the parameter values of a specific function (i.e., in terms of its mathematical expression), GP rather focuses in predicting a distribution of functions, characterized by a mean (i.e., the most probable function = the prediction) and a variance (i.e., the uncertainty of the prediction), which are consistent with the available (or newly introduced) data. Additionally, another interesting characteristic of GP, which could be exploited in a hybrid modeling framework, is the possibility to introduce prior knowledge through the prior mean and covariance (or kernel) of this method. Concretely, GP predicts a posterior distribution of functions (characterized by a posterior mean and covariance), based on a prior distribution of functions (also characterized by a prior mean and covariance) and available training data. The prior distribution, corresponding to a large number of candidate functions for a given problem, is progressively updated with the received training points and finally results in a posterior distribution. The latter presents therefore lower uncertainties in the areas where training points were available. Several references provide complete explanation of the GP method as well as its mathematical components [26, 34, 73, 77].

The implementation of GP as part of a hybrid modeling scheme is quite recent. Indeed, most of the applications found in the literature were published during the last five years. In several works, GP was used as a way to correct the residuals between the observations and the predictions obtained via a WB model [17, 19, 45, 57, 69]. Some works also used the WB predictions as additional inputs for GP models [69, 79]. An interesting work is the one of [74] which investigated the impact of different levels of greyness of hybrid modeling for bioprocess predictive modeling. Concretely, three hybrid models with different amounts of incorporated

kinetic knowledge were compared. The GP model was embedded inside the kinetic model (WB) to describe the unknown mechanisms. All three hybrid models displayed better performance than the pure kinetic model, especially the model with moderate kinetic information. Indeed, the model with low kinetic information was more prone to overfitting, while the one with more kinetic information incorporated incorrect information.

All the aforementioned works employed basic prior means and covariances for GP models. Basic prior means include zero or constant mean, while basic covariances (or kernels) include square exponential or RBF (radial basis function) kernel, Matern kernel or combinations of basic kernels [66, 73]. However, more sophisticated prior means or covariances were developed in the literature to provide more physically realistic predictions. For example, in [62], the considered prior mean was a closed-form solution of the kinematics model (WB model) for the modeling of industrial robot kinematics. In other works, the covariance function was replaced by deep neural networks which extract hidden deep features or learn an adaptive covariance function based on both data and physics constraints [12, 26, 54]. In [68], the prior knowledge was integrated by deriving the mean and covariance function from previous data. In [26] and [16], composite kernels were employed, consisting in linear combinations or products of basic kernels. These complex kernels can result in better accuracy while limiting the number of required training data. However, they consequently contain more parameters and are less interpretable. Finally, different works developed specific kernels to improve extrapolation capabilities of GP. In [91], spectral mixture kernels were developed for automatic data pattern discovery and extrapolation. In [70], evolutionary GP enables to automatically discover some free-form symbolic bases describing the data reasonably well.

Another way to produce more realistic predictions consists in building constrained GP. The constraints apply to GP parameters and include for example, boundary values, inequalities, convexity or monotonicity [5, 21, 39, 46, 53, 67, 82, 85, 87]. In particular, the work in [32] demonstrated that adding boundary constraints to a physics-based GP model yields more accurate and stable solution inference than in absence of constraints. Finally, a parallel hybrid configuration is found in [27] where physics-based, process state and data-driven modeling are combined via a hidden Markov model. Other examples of complex hybrid structures are also reported [18, 42, 51, 52, 55].

5.2.3 Hybrid modeling and Gaussian Processes in polymer engineering

5.2.3.1 Gaussian Processes-based hybrid models in polymer engineering

In the polymer engineering field, the application of GP-based hybrid models remains limited. An example is the work in [8] where different methods for incorporating simple theory into GP were investigated for the prediction of polymer radius of gyration in different solvent qualities and for different chain lengths. These methods include different serial and parallel configurations as well as different operations to combine WB and BB predictions.

5.2.3.2 Hybrid models in polymer engineering

Inversely, the applications of either hybrid modeling or GP in polymer engineering can be more easily identified. On the one hand, hybrid modeling has been employed mainly for process modeling, optimization and control. A typical category of such applications concerns the use of BB models to predict the errors made by WB models. In [84], the residuals of a simplified mechanistic model that does not consider the gel effect were predicted with a stacked recurrent neural network for the modeling and control of a batch polymerization reactor (prediction of conversion, number- and weight-average molecular weights). In [24], a fundamental population balance model, describing the particle size distribution in emulsion polymerization, was combined with a partial least squares model. In particular, the latter was used to predict the residuals between the WB model and the experimental measurements. A similar study was implemented in [37] to control the particle size distribution in emulsion polymerization. In [35], an industrial polyethylene process was modeled via a hybrid approach. In particular, three feed-forward artificial neural networks (ANN) were employed to adjust the predictions of the mechanistic model, and the corrected predictions were then used to calculate a more precise molecular weight distribution.

Another type of application uses the BB models to predict some inputs of the WB models. In [6], a hybrid modeling approach was implemented to describe the cellulose acetate-based historical film degradation process. Two chemometric PLS models (BB) were employed to predict two additional inputs for the WB model, describing the experimental degradation state of given films, additionally to the storage conditions. In [7], a hybrid model, combining material balances and two ANN, was implemented for the prediction of the evolution of the monomer conversion and the average molecular weight in semi-continuous emulsion polymerization with long-chain branching kinetics. Concretely, the two ANN predicted the polymerisation rate and the instantaneous average molecular weight from process variables. These predictions were then used by the WB model for the integration of the material balances.

Inversely, WB models can be used to predict additional inputs for BB models. In [81], the polymer melt index (MI) of an industrial polymerization process was predicted via a hybrid model in which the BB model received process data (e.g., temperature, pressure) and the WB model predictions as inputs to predict the MI. In particular, the WB model included the mass and energy balances as well as the reaction kinetics and thermodynamics (which are well known for radical bulk polymerization) and predicted the molecular weights, polymerization rates and degree of polymerization based on process data. Different BB models were compared including PLS, DT (decision tree), SVM (support vector machines), ANN and GP. The prediction accuracy and generalizability was better for the hybrid model than for the pure BB model trained on process data only.

5.2.3.3 Gaussian Processes in polymer engineering

Concerning the GP utilization in polymer engineering, applications can be mainly identified again in process modeling, optimization and control but also in product engineering. For example, GP have been mostly applied for predicting glass transition temperature [9, 38, 71, 83, 96, 97] and MI [11, 14, 31, 49, 50]. Other reported physical properties include polymer dielectric properties (dielectric constant, breakdown strength, permittivity, energy density) [13, 80, 100], refrac-

tive index [48], electrical bandgap [9, 65], viscosity [47], gas permeability [100], density, melting temperature, coefficient of thermal expansion and Young's modulus [32, 36]. To accelerate design and discovery, the team of Ramprasad and Kim also developed the PolymerGenome platform which uses GP to predict diverse polymer properties (such as electronic, mechanical or dielectric properties) from polymer fingerprints [22, 38, 41, 71, 72, 100].

Otherwise, GP have also been employed to investigate imperfections in polymer-based products caused by manufacturing in industrial processes (e.g., injection moulding, production of glass fiber reinforced polymers) or due to severe conditions (repeated fatigue loads, humid environment) [15, 33, 50, 64, 98, 99].

In industrial plants, the MI is the main predicted property using GP [11, 14, 31, 49, 50]. The control of MI in a polymerization process is crucial as it relates to the grade of the produced polymer. It is typically analyzed off-line, a procedure that may cause important delays between the production and the characterization of the product and, as such, in the quality control process. This often results in the production of non-negligible quantities of out-of-specifications products. Online measurement tools are expensive and need important maintenance efforts. Unlike WB models that are time-consuming to develop, as they require deep understanding of the rheological and physico-chemical behavior of the macromolecular system, ML methods have been increasingly adopted as soft sensors for the agile manufacturing of products with changing specifications. Additionally, to account for batch process characteristics, some works deployed particular structures of GP. For example, [94] developed GP with multi-kernels for state estimation and quality prediction of nonlinear batch processes with multiple operating phases and between-phase transient dynamics. In [63], a mixture model of GP was implemented for a more accurate modeling of the behavior of batch processes with data having different variance structures throughout the duration of the batch.

Another application in which GP have been used is the characterization of polymer microstructures as it is particularly challenging, while the microstructure of polymers is also highly correlated to a series of polymer properties. For example, GP were used to predict Minkowski functionals of polymer blends from scattering data (obtained from light, X-ray or neutron techniques) [40]. Minkowski functionals contain key characteristics of morphological image analysis such as total volume occupied by one of the phases, combined surface area of the interfaces between the phases, average curvature of the interphases or connectivity between two phases.

Finally, a review of GP for materials and molecules more generally is provided in [20]. In the aforementioned applications, GP were selected for similar reasons, namely their ability to provide the uncertainties related to the predictions as well as to account for noise existing in experimental and simulation data. Besides, GP can perform well in presence of small data sets and are robust to overfitting [30, 47, 75, 78]. Data sets generally contain several tens or hundreds of samples. In the industrial production of multiple-grade products, GP advantages are the possibility to be trained by optimizing its parameters automatically and its ability to address well high nonlinearity of the process data in each operation mode. Another benefit is that one single global GP model is sufficient to describe all the process characteristics and operation modes without additional information of the process [31, 50].

5.3 Data sets and methods

5.3.1 Data sets

5.3.1.1 Literature experimental data

The experimental data of several reported works, on the polymerization of styrene, were exploited to build the hybrid models presented in this chapter. These data are summarized in Table 5.2. The column "Data" refers to the number of available data, corresponding to different reaction times.

Table 5.2: Literature-extracted experimental data on the polymerization of styrene.

Case	Reference	Initiator (g)	Styrene (g)	Solvent (g)	T (°C)	Data
L1	Florez et al. [28]	4.35 [a]	95.65	0	85	19
					90	19
					120	19
L2	Villalobos et al. [89]	0.29 [a]	99.71	0	90	11
L3	Kotoulas et al. [44]	0.4 [b]	99.6	0	120	13
					130	12
					140	12
					150	11
L4	Kotoulas et al. [44]	0	100	0	160	12
					170	11
					180	11
L5	Vicevic [88]	1.37 [a]	50.56	48.07 [c]	80	19
		1.36 [a]	82.49	16.15 [c]	90	14

[a] *BPO: benzoyl peroxide.* [b] *DCP: dicumyl peroxide.* [c] *Toluene.*

5.3.1.2 Additional experimental data

Additional data was generated experimentally to measure the evolution of styrene conversion (in absence or presence of GTR), at various temperatures and different reactant proportions. The following reactants are considered for the styrene-GTR radical graft polymerization, similar to the previous studies (cf. Chapter 4):

- Monomer: styrene Reagent-Plus®, with a purity ≥99% and stabilized with 4-tert-butylcatechol (Sigma-Aldrich)
- Initiator: BPO Luperox® A75, 75%, remainder water (Sigma-Aldrich)
- GTR (DeltaGom France): characteristics given in [28, 29]

All the reactants were used without further purification, for the same environmental- and cost-related arguments that were put forward in Chapter 4. As for the experimental procedure,

each polymerization reaction was performed in a beaker, containing a total of 1.5g of reactants (BPO, styrene and GTR). Concretely, GTR was first introduced in the beaker, then a mixture of styrene/BPO, prepared beforehand, was introduced onto the GTR. The beaker was subsequently sealed and introduced in an oil bath, which was preheated at the target temperature. At the end of the reaction, the beaker was directly introduced in a cold water bath for cooling.

To measure the styrene conversion, the beaker was unsealed and placed inside a vacuum oven at 40°C to eliminate the unreacted styrene at the end of the reaction. Note that beakers were used instead of test tubes, as was the case in the previous study, to facilitate the evaporation of styrene in the vacuum oven via the larger evaporation surface. Finally, the styrene conversion was calculated as follows:

$$X_{styrene} = \frac{m_{i,styrene} - m_{f,styrene}}{m_{i,styrene}} \tag{5.1}$$

where $m_{i,styrene}$ denotes the mass of styrene that was introduced in the beaker, before the reaction, and $m_{f,styrene}$ is the recorded mass loss during the vacuum evaporation. Note that, any eventual mass loss that occurred during the polymerization reaction (e.g., due to eventual styrene evaporation through the seal) did not exceed 0.2g in all cases and was taken into consideration in the value of $m_{i,styrene}$ in Equation 5.1.

The tested experimental conditions are summarized in Table 5.3. In particular, two categories of experiments were conducted, namely without GTR (cases E1, E2 and E3 at 90°C, 100°C and 110°C, respectively) and in the presence of GTR (cases E4 and E5 at 90°C and 100°C, respectively). For the second category, a ratio of GTR/(GTR+styrene) equal to 10%wt (theoretical value, ± 2%wt) was considered. For both categories, the ratio BPO/styrene was set to 5.5%wt (theoretical value, ± 1%wt). The detailed collected data is available in Appendix 5.5. Note that some experiments were repeated several times to verify the reproducibility of the experimental protocol. The values shown in Table 5.3 correspond to the average experimental values. The results of the experiments under higher amounts of GTR (i.e., 30%wt) led to very low styrene conversion values (below 10% in all cases) and to comparable repetition variations. Accordingly, they were considered as less reliable and omitted from the considered data in this study.

Table 5.3: Additional experimental data, generated during this study.

Case	**BPO (g)**	**Styrene (g)**	**GTR (g)**	**GTR/ (GTR+styrene) (%)**	**BPO/ styrene (%)**	**T (°C)**	**Data**
E1	5.36	94.64	0	0	5.7	90	5
E2	5.24	94.75	0	0	5.5	100	4
E3	5.14	94.86	0	0	5.4	110	4
E3	4.80	85.51	9.69	10.2	5.6	90	5
E4	4.80	85.50	9.70	10.2	5.6	100	5

5.3.2 Methods

5.3.2.1 WB model

The knowledge-based model developed during a precedent thesis was considered for the hybrid modeling in this work [29, 58]. This model consists in a generalized kinetic scheme describing the course of the styrene-GTR radical graft polymerization, under different operating conditions (GTR, peroxide initiator contents and reaction temperatures). The model is general in the sense that it describes both the case of classical styrene radical homopolymerization (i.e., in absence of GTR) and the one of styrene radical polymerization in presence of GTR. In particular, the switching from one case to another can be done simply by adjusting the quantity of GTR in the general model.

The model relies on the classical styrene homopolymerization kinetics and considers a set of chemical reactions that are triggered by the presence of GTR particles, as observed and reported in several works of the literature [58]. The polymerization kinetic mechanism is provided in Appendix 5.5 and further detailed in [58]. It includes chemical initiation, thermal initiation, propagation, transfer, termination of the classical styrene homopolymerization. In addition, the role of carbon black, which is a common reinforcing agent of GTR, is included in the model in terms of the observed deviation of the behavior of the system with respect to that of the classical homopolymerization. Carbon black contains multiple highly-reactive functional groups and double bonds causing grafting reactions, catalyzed initiator decomposition as well as radical deactivation. Accordingly, these reactions induced by carbon black were considered in the kinetic model as well.

To develop the knowledge-based model, the rates of production of the species present in the polymerization system were first expressed based on the kinetic mechanism. In particular, two categories of species were considered, namely macromolecular and non-macromolecular species. The second category is composed of species with small and/or fixed chain length during the polymerization, such as styrene monomer, initiator, primary radicals, GTR reaction sites and carbon black. Inversely, in the first category, different types of macromolecular species (free/grafted/reduced-activity radicals, free/grafted polymer chains) with varying chain lengths were included, thus leading to a system of ordinary differential equations (ODE) (cf. Equations 5.5 and 5.6). For these macromolecular species, in order to reduce the size of the system of ODE for computational reasons, the method of moments was implemented [43]. This statistical method allows to express the average molecular properties of the macromolecular species (e.g., number and mass average molar mass) as a function of their leading moments. The moment of order k for the free radicals R_n can be described as follows:

$$\lambda_k = \sum_{n=1}^{\infty} n^k R_n \tag{5.2}$$

where n represents the chain length.

For example, the method of moments enables to express the rate function of free radicals r_{R_n} (cf. Equation 5.3) as the rate function of their leading moments r_{λ_k} (cf. Equation 5.4). The expressions of all the rate functions for both macromolecular species (leading moments) and non-macromolecular species are further detailed in [58].

$$
\begin{aligned}
r_{R_n} = & \left(k_I \cdot [PR] \cdot [M] + k_{fm} \cdot [M] \cdot \sum_{k=1}^{\infty} [R_k] \right) \cdot \delta(n-1) \\
& + k_A \cdot [AR] \cdot [M] \cdot \delta(n-3) + k_B \cdot [MR] \cdot [M] \cdot \delta(n-2) \\
& + k_p \cdot [M] \cdot ([R_{n-1}] - [R_n]) - (k_{fm} \cdot [M] + k_{fAH} \cdot [AH]) \cdot [R_n] \\
& + k_s \cdot [PR] \cdot \sum_{k=n+1}^{\infty} [D_k] - (k_{tc} + k_{td}) \cdot [R_n] \cdot \sum_{k=1}^{\infty} [R_k] \\
& - k_{tPR} \cdot [PR] \cdot [R_n] + \left(k_{fmG} \cdot [M] \cdot \sum_{k=1}^{\infty} [GR_k] \right) \cdot \delta(n-1) \\
& - k_{fG} \cdot [G] \cdot [R_n] + k_{sG} \cdot [PR] \cdot \sum_{k=n+1}^{\infty} [GD_k] \\
& - (k_{tcG} + k_{tdG}) \cdot [R_n] \cdot \sum_{k=1}^{\infty} [GR_k] - k_{dea} \cdot [CB] \cdot [R_n] \\
& - \alpha \cdot [R_n] \cdot \left((k_{tc} + k_{td}) \cdot \sum_{k=1}^{\infty} [P_k] + (k_{tcG} + k_{tdG}) \cdot \sum_{k=1}^{\infty} [GP_k] \right)
\end{aligned}
\tag{5.3}
$$

$$
\begin{aligned}
r_{\lambda_k} = & \, k_I \cdot [PR] \cdot [M] + k_{fm} \cdot [M] \cdot \lambda_0 \\
& + 3^k \cdot k_A \cdot [AR] \cdot [M] + 2^k \cdot k_B \cdot [MR] \cdot [M] \\
& + k_p \cdot [M] \cdot \left(\sum_{r=0}^{k} \binom{k}{r} \lambda_r - \lambda_k \right) - (k_{fm} \cdot [M] + k_{fAH} \cdot [AH]) \cdot \lambda_k \\
& + k_s \cdot [PR] \cdot T_1 - (k_{tc} + k_{td}) \cdot \lambda_k \cdot \lambda_0 - k_{tPR} \cdot [PR] \cdot \lambda_k \\
& + k_{fmG} \cdot [M] \cdot \nu_0 - k_{fG} \cdot [G] \cdot \lambda_k + k_{sG} \cdot [PR] \cdot T_{1G} - (k_{tcG} + k_{tdG}) \cdot \lambda_k \cdot \nu_0 \\
& - k_{dea} \cdot [CB] \cdot \lambda_k - \alpha \cdot \lambda_k \cdot ((k_{tc} + k_{td}) \cdot \theta_0 + (k_{tcG} + k_{tdG}) \cdot \omega_0)
\end{aligned}
\tag{5.4}
$$

The amount of available reaction sites on the accessible internal and external surface of the GTR particles, denoted as G, was also considered in the model. The initial concentration, $[G]_0$, was based on the quantity of double bonds, an efficiency factor reflecting the available reaction sites (i.e., prone to react) and an additional term to describe the coverage or saturation of these reaction sites as observed in [28]. The quantity of double bonds was measured experimentally and estimated theoretically on the basis of the information provided by the supplier of the GTR used in [58]. This includes the type and weight fraction of the elastomers constituting GTR.

The diffusion controlled phenomena (i.e., gel-, glass- and cage- effects) affecting termination, propagation and initiation steps were also included in the knowledge-based model based on the work of Marten and Hamielec [56]. Inversely, mass transfer and diffusion phenomena within GTR were considered negligible to simplify the model in view of the lack of accurate measurements to determine the associated parameters. Further more specific assumptions on the model can be found in [58] (e.g., quasi steady state assumption for the determination of the concentration of primary radical species).

Finally, the reactor design equations for this batch system could be fully expressed for all

species (i.e., non-macromolecular species and moments of the macromolecular species) as follows:

$$\frac{dn_S}{dt} = r_S \cdot V \tag{5.5}$$

where S corresponds to the different species and V is the volume of the reacting mixture, which is defined by the following equation:

$$\frac{dV}{dt} = r_M \cdot \left(\frac{1}{d_M} - \frac{1}{d_P}\right) \cdot M_0 \cdot V \tag{5.6}$$

M_0 being the molecular weight of styrene and d_M and d_P denoting the respective densities of styrene and PS in $g.mol^{-1}$, calculated by the following expressions [44]:

$$d_m = 9.236 \cdot 10^{-1} - 0.887 \cdot T(^\circ C) \quad ; \quad d_p = 1.085 - 0.605 \cdot T(^\circ C) \tag{5.7}$$

The model was calibrated and validated on experimental data from the literature (polymerization in absence of GTR) and additionally-generated experimental data during that study (polymerization in the presence of GTR). More precisely, the kinetic rate constants were estimated in a two-step procedure. First, the constants from the pure homopolymerization were estimated on the basis of homopolymerization data. Note that some constant values were taken from the literature and that some commonly adopted assumptions were considered (e.g., $k_A = k_B = k_I = k_p$). In a second step, the constants of the styrene polymerization in presence of GTR were estimated based on the corresponding data (i.e., polymerization in presence of GTR). In this step, the kinetic model was reduced by keeping only the reactions directly related to monomer conversion evolution due to the lack of relevant data describing the whole model. The reduced kinetic model is displayed in blue in Appendix 5.5. In addition, some parameters were set to zero due to their low impact when polymerizations at relatively high temperatures for long duration are not implemented. The final set of estimated parameters for the WB model used in the hybrid model is presented in Appendix 5.5. Note that in the present work, a value of 0.83 was considered for the initiator efficiency during the application of the WB model under the considered experimental conditions in Table 5.3. The whole system of ODEs was numerically solved on Matlab via a variable-step, variable-order (VSVO) solver based on the numerical differentiation formulas (NDFs) of orders 1 to 5, implemented by the internal function "ode15s".

5.3.2.2 BB model

Gaussian Processes (GP) were employed for the BB part of the hybrid modeling scheme. In particular, the estimator *GaussianProcessRegressor* from Python's Scikit-Learn library was employed with the options described in Table 5.4.

A zero mean was considered as the prior mean via the option *normalize_y* set to False. As for the kernel, the RBF (a.k.a. square exponential kernel), was chosen. Indeed, some kernels such as the constant, the exp-sine-squared and the dot product kernels could easily be dismissed because they are not consistent with the known shape of styrene conversion evolution (not constant, not periodic and not invariant to rotation respectively). Besides, the RBF kernel was preferred over the Matern and the rational quadratic kernels which are more complex as they correspond to a generalization of the RBF kernel and to an infinity sum of RBF kernel with different length

Table 5.4: Gaussian Processes options.

Option type	Option values
kernel	1.0*RBF(length_scale=1.0, length_scale_bounds=(1e-5, 1e5))
alpha	1e-2, 1e-3 or 1e-4
optimizer	fmin_l_bfgs_b
n_restarts_optimizer	10
normalize_y	False (prior mean = zero)

scale, respectively. More generally, the RBF kernel is widely used for its ability in representing very smooth functions and is expressed as follows:

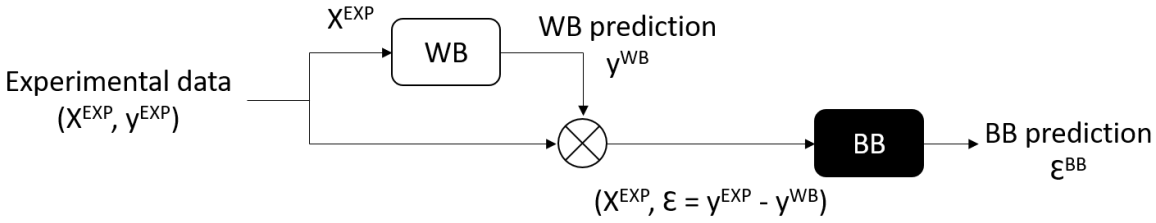
$$k(x_i,x_j) = exp(-\frac{d(x_i,x_j)^2}{2l^2}) \tag{5.8}$$

where $d(x_i,x_j)$ is the Euclidean distance between the points x_i and x_j and l is the length-scale.

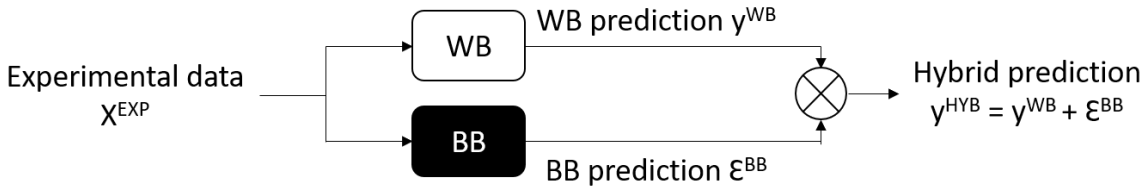
The option *n_restarts_optimizer* corresponds to the number of restarts of the considered optimizer (L-BFGS-B algorithm) to identify the best parameters (here the length-scale l) of the considered RBF kernel, maximizing the log-marginal likelihood. The parameter α, corresponding to variance of additional Gaussian measurement noise on the training observations, was adjusted manually, within the range [1e-2-1e-4], for each case study to obtain more physically realistic predictions (i.e., avoid wavy predictions). This point requires further improvement within a future study as alternative approaches to include the data uncertainties can be envisioned.

5.3.2.3 Hybrid model

The WB and BB models previously described are combined to build a hybrid model, as shown in Figure 5.2. The WB model is composed of well established equations and knowledge about the system, on the basis of the extensively studied polymerization kinetics of styrene. The role of the BB model is to represent the poorly understood phenomena considering the presence of GTR, as well as eventual experimental uncertainties, as analyzed previously. Figure 5.2a describes the fitting procedure on the basis of a serial configuration. The BB model is trained on a data set composed of the experimental conditions X^{EXP} and the residuals between the experimental measurements y^{EXP} and the WB model predictions y^{WB}. This data set is divided into training and test sets following a ratio of 70%/30% for the literature cases, where more data was available than in the additional experimental data sets. This configuration provides an indirect manner of replacing the prior of the GP component with the model predictions (i.e., since the input to the GP, corresponding here to the difference between the predictions of WB and experimental data, has a prior set to zero, it is *a priori* considered that the WB model predictions are the starting point of the training). As for Figure 5.2b, it corresponds to the application of the trained hybrid model to any experimental conditions X^{EXP}. The configuration in this case is parallel: both the WB and BB models receive as inputs the experimental conditions X^{EXP}, then their predictions y^{WB} and ϵ^{BB} are added to form the final hybrid prediction y^{HYB}. The performance of the WB, BB and hybrid models were compared based on the mean absolute error (MAE) values.



(a) Model fitting



(b) Model application

Figure 5.2: Hybrid model fitting and application.

As a preliminary work, this chapter focuses only on the previously presented hybrid structure, however other structures could be envisioned as later improvements. In fact, as the WB model was already available and displayed predictions more or less accurate depending on experimental conditions, the adopted cooperative parallel structure is the simplest to implement at first. The serial configuration BB/WB would require a re-development of the WB model that would include parts of the WB model predicted by the BB model. As for the WB/BB serial configuration, an idea is to determine the Gaussian Processes kernel and prior mean from the WB model. Another idea consists in adding constraints on the Gaussian Processes model (i.e., styrene conversion bounded to [0:1] and increasing trend). However, in both cases, complex mathematical developments are necessary but they are considered as future steps of this work (cf. conclusions and perspectives).

Separated hybrid models were trained for each of the experimental cases presented in Tables 5.2 and 5.3 in order to build simple models (i.e., with one single input: polymerization reaction time) that facilitate the analysis of the results. The consideration of all the experimental cases (and therefore more inputs: reactant contents, polymerization reaction temperature) to build one single hybrid model can then be considered as a future improvement.

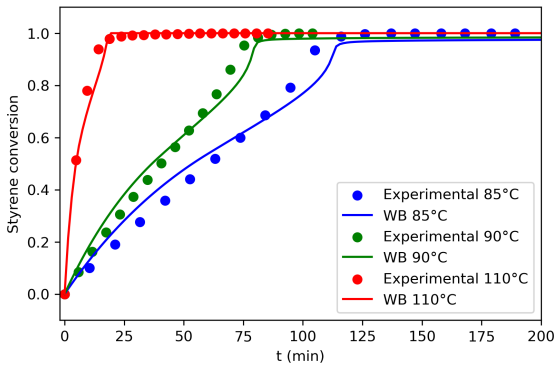
The ML and hybrid modeling were implemented using the Scikit-learn library v1.0.2 of Python v3.9.12 [66], while Matlab was used for the preliminary development of the WB model and the determination of the WB model predictions.

5.4 Preliminary results

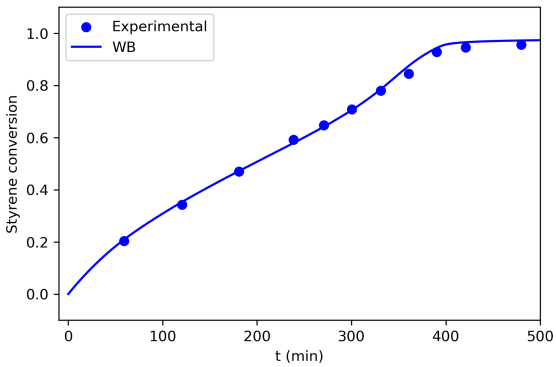
5.4.1 WB modeling

Initially, the WB model was applied independently under the same conditions (i.e., reactants contents and type, reaction temperature) as those used to generate the experimental data. The

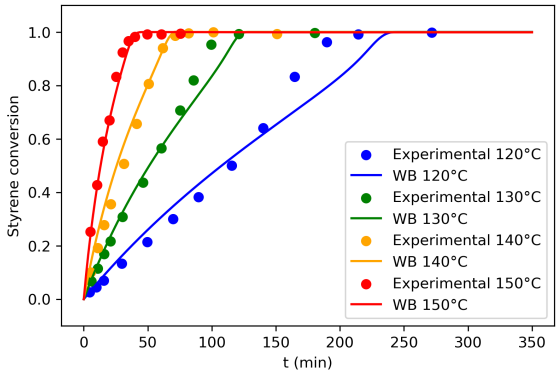
comparison of the WB model predictions and the experimental data is displayed in Figures 5.3 and 5.4, for the literature and the additional data cases, respectively.



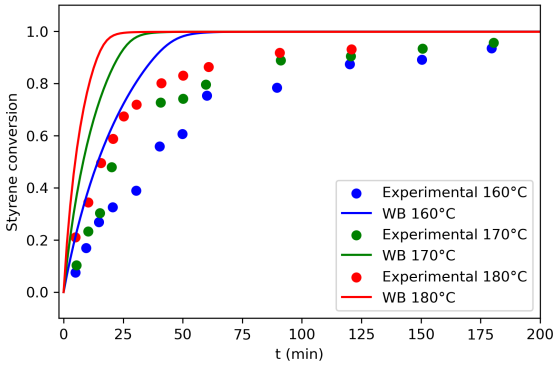
(a) L1 (Florez et al.)



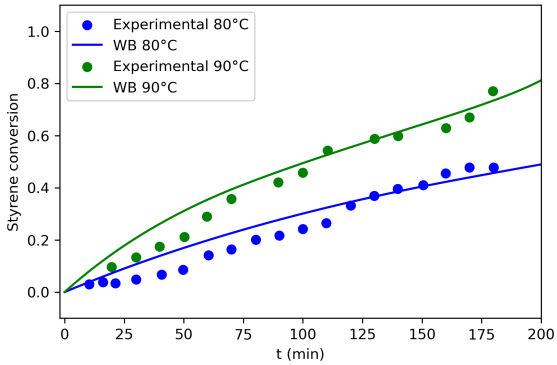
(b) L2 (Villalobos et al.)



(c) L3 (Kotoulas et al.)



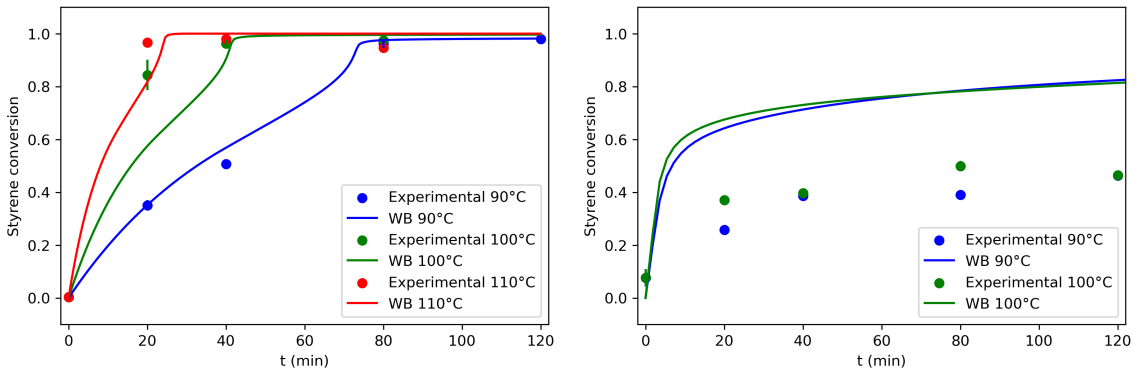
(d) L4 (Kotoulas et al.)



(e) L5 (Vicevic et al.)

Figure 5.3: Comparison of experimental data and WB model for cases L1 to L5.

It can be observed that the WB model provides predictions close to the experimental observations for most of the literature cases such as in L2 case and in L1/L3 cases for higher temperatures. However, for other literature cases (i.e., L4, L5 and L1/L3 for lower temperatures), some deviations are observed. These deviations can be attributed to several reasons. A closer examination reveals that the WB model presents the highest deviations for the cases of pure thermal initiation (cf. L4) as well as for the case of solution polymerization (cf. L5). Accordingly, a higher uncertainty in the estimation of the values of the relevant kinetic rate con-



(a) E1/E2/E3 (without GTR)

(b) E4/E5 (with GTR)

Figure 5.4: Comparison of experimental data and WB model for cases E1 to E5.

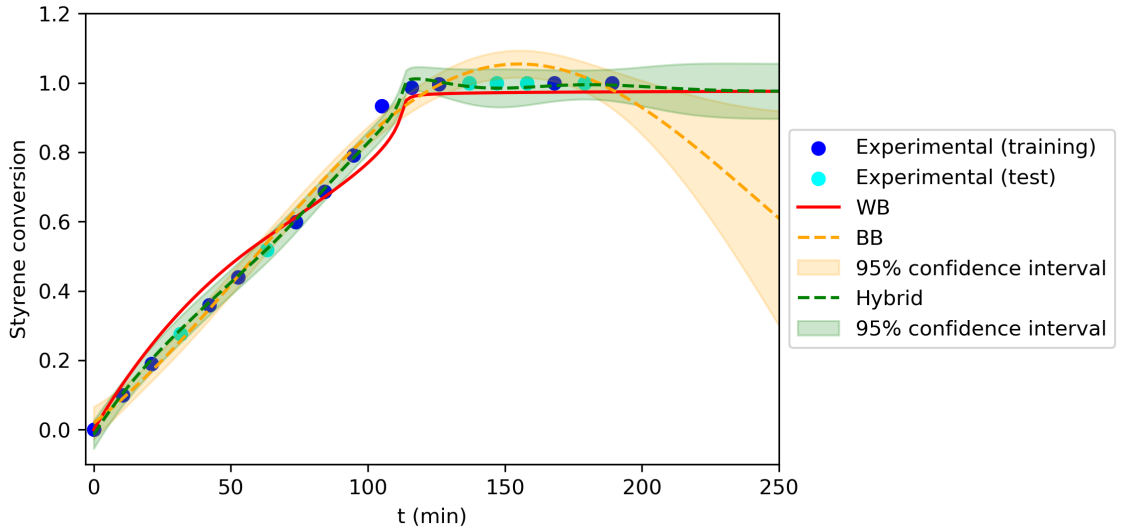
stants (i.e., those related to the thermal initiation and the transfer to solvent mechanisms) could be related to these deviations, as the model would fail to describe accurately these phenomena, as in the work of [74] (see Section 5.2.2). The description of the diffusion limitations, through the respective models of gel- glass- and cage-effects, under these conditions could also increase the prediction uncertainties. These examples highlight the limitations of pure WB models when the system's behavior is not accurately represented, hence the interest in combining these WB models with BB models to better describe the effects not captured by the the former, under experimental uncertainties.

Regarding the cases E1 to E5 (i.e., those with additional experimental data), deviations are also observable, in particular in presence of GTR (cases E4 and E5). The unknown composition and complex structure of GTR (e.g., weight fraction of elastomers, particle size, additives) as well as its complex interactions with the reactive mixture could cause these deviations. For example, the quantity of double bonds in GTR in the kinetic model was kept equal to the value determined in [58], which could no longer be applicable in the present case. Again, the WB model includes a set of reactions occurring due to the presence of GTR, which might not fully represent the occurring phenomena, and some assumptions and simplifications were made (cf. Section 5.3.2.1), which might no longer be applicable. In general, a possible solution is to use hybrid models where the WB part models only the "fully-understood" mechanisms and the BB part predicts the residuals with experimental data. Otherwise, the WB model can be further improved by better describing the missing mechanisms and providing further data to determine the parameters. However, the phenomena are very complex and time-consuming to describe and there will still be uncertainties in the experimental data as well as missing phenomena due to the complex composition and structure of GTR.

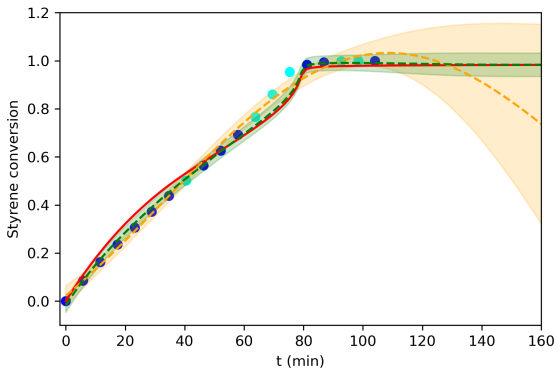
5.4.2 Comparison of WB, BB and hybrid modeling

In this section, the performances of WB, BB and hybrid models are compared for the different data cases. The entire data was used to train and validate the different models, following a training/test ratio of 70%/30% for each subcase, except for those based on the additional experimental data (i.e., data generated during the present study), in which all available data was attributed to the training set, due to their limited amount. The graphical comparisons of

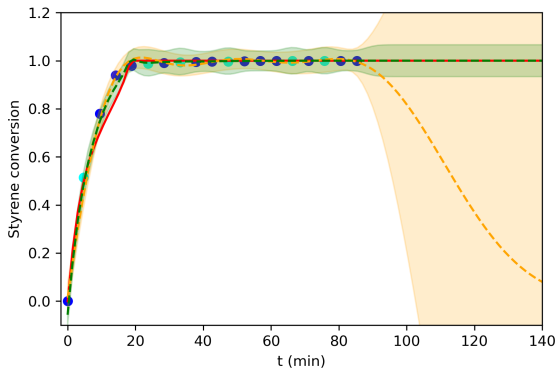
WB, BB and hybrid models are presented for a given split in Figures 5.5 to 5.9, for the literature cases, and in Figures 5.10 and 5.11, for the cases based on the additional experimental data.



(a) 85°C



(b) 90°C



(c) 120°C

Figure 5.5: Comparison of WB, BB and hybrid models - Case L1 (Florez et al.).

Concerning the model predictions within the time range defined by the training data, both the hybrid and the BB model present a good fit to the observations in all cases in comparison with the WB model. However, given that the hybrid model has a physical component, the shape of the hybrid predictions is more realistic than for the pure BB model in most cases. Indeed, the decrease in the styrene conversion, which is observed in Figures 5.10b and c, Figure 5.5a or in Figure 5.7, is physically unrealistic. This behavior is an artifact of the purely data-driven character of the BB model and is mainly observed in areas of a lower density of training data. This highlights one of the main limitations of pure BB models, whose predictions are sometimes inconsistent with the physical reality, and enhances the interest in combining them with knowledge-driven models.

Besides, an exception of a similar decreasing behavior is also observed for the hybrid model, in Figure 5.7a, mainly attributed again to the lack of training data but also to the higher residual between the observations and the WB model predictions (i.e., to the lower significance of the

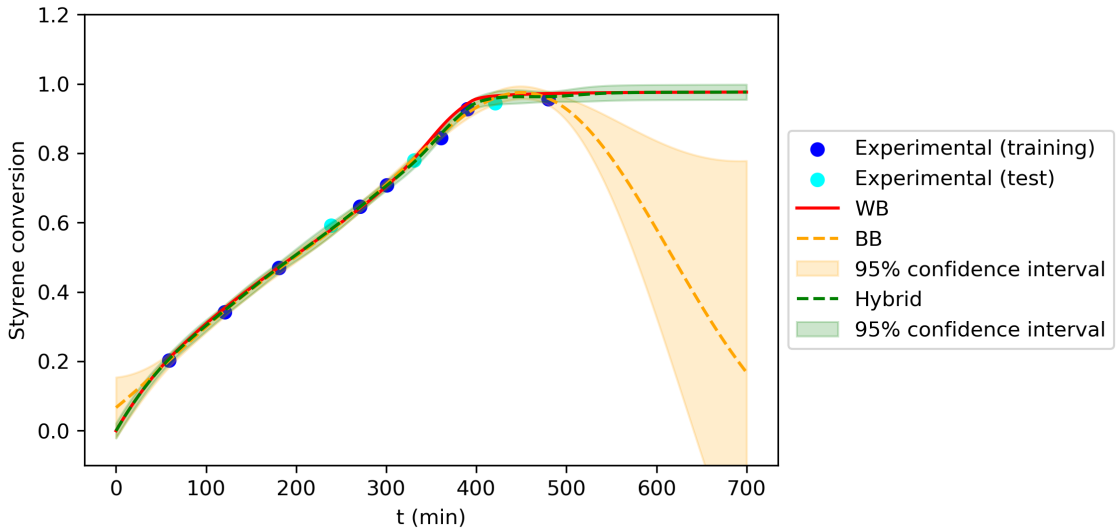
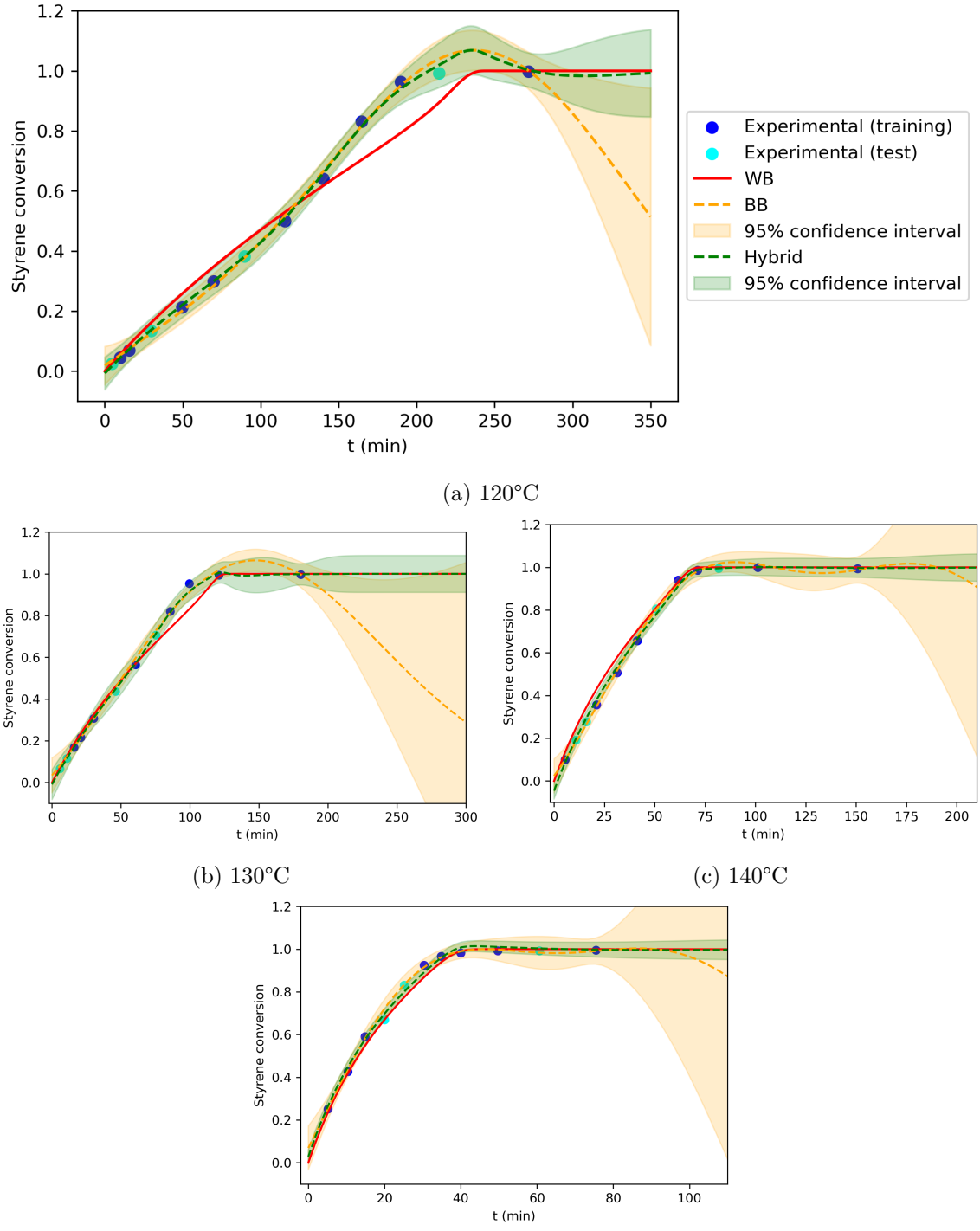


Figure 5.6: Comparison of WB, BB and hybrid models - Case L2 (Villalobos et al).

contribution of the knowledge-based model with respect to the observations). In fact, in a few cases, the BB model seems to outperform the hybrid model (within the range of available data), especially when the residual between the observations and the WB model is higher. This is for example the case in Figure 5.5b, where the corresponding area contains no training data and presents a more significant deviation between the observations and the WB model predictions. This is to be compared with Figure 5.5a which presents a similar area with no training data but a low deviation between the observations and the WB model. In this case, the hybrid model performes better than the BB model even within this area. These observations seem consistent with the literature concerning the performance of the serial and parallel configurations, depending on the accuracy of the WB model (see Section 5.2.1). Indeed, the serial configuration generally performs better than the parallel configuration under the condition that the WB is accurate enough, otherwise the BB model can perform better than the hybrid model. The case with GTR (cf. Figure 5.11) is another more obvious example of this situation. As the WB model predictions are very far from the observations, the serial hybrid model seems to not fit the data well.

In case of predictions outside the range of available data (i.e., in extrapolation), it can be observed that the BB and the hybrid models converge towards the defined prior mean in the GP model, namely the zero mean and the WB model, respectively. This makes the hybrid model significantly more reliable when predictions outside the training range are required. However, here again, the WB needs to be accurate enough, at least in the case of the serial configuration.

In fact, this reveals another major benefit of the use of similar hybrid modeling structures. In cases where the data generated by a running process diverge from the expected trend (e.g., due to the aforementioned reasons), as predicted by an established WB model, there is no option to correct the predictions other than to estimate from scratch the model parameters. This is highly impractical and time-consuming and can render useless the existence of the model for the specific case. Should a BB model be employed instead, the same issues related to the available data (i.e., the data used also by the WB model) and their capacity to describe this newly observed deviation would arise. In the specific case of a GP, the model predictions would converge to



(a) 120°C

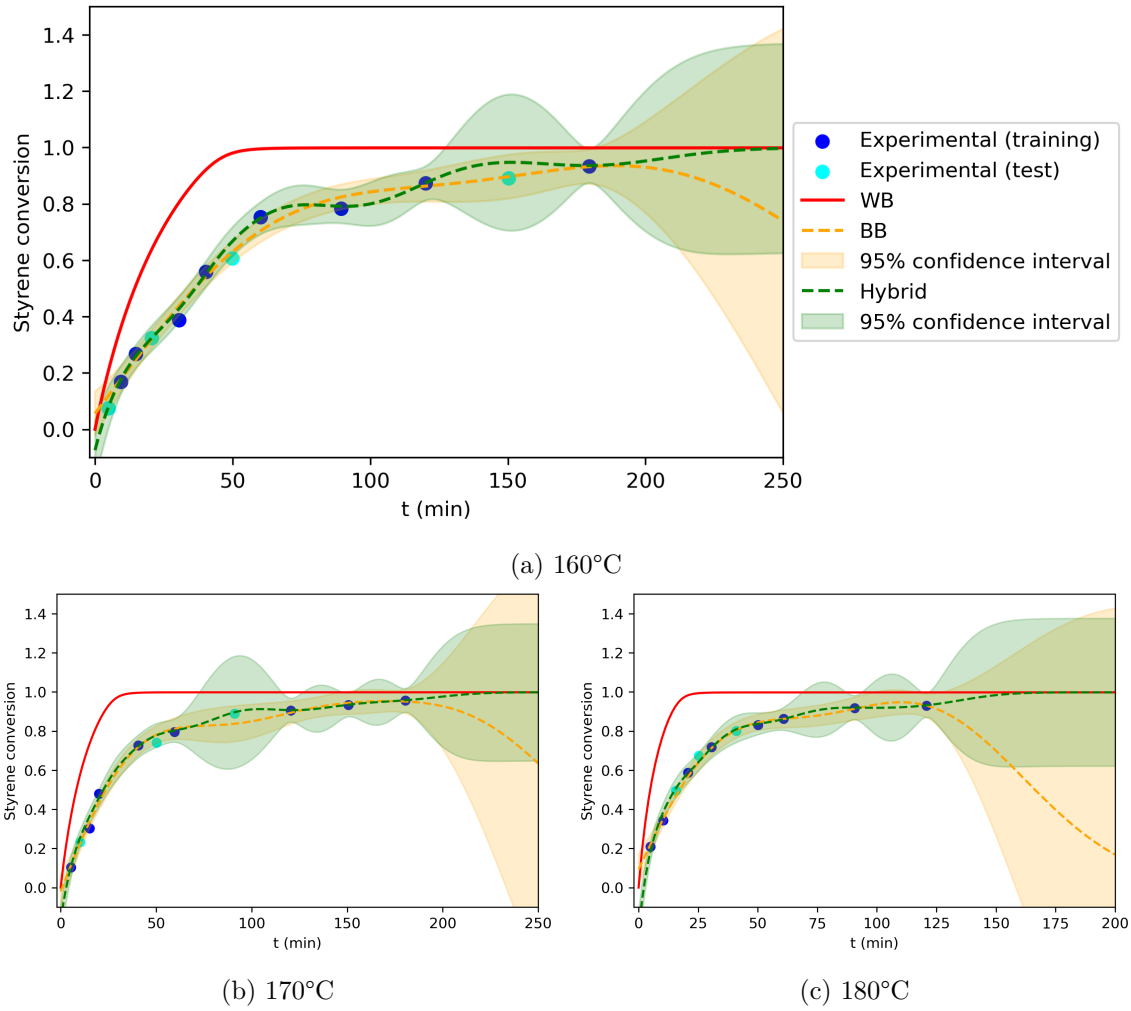
(b) 130°C

(c) 140°C

(d) 150°C

Figure 5.7: Comparison of WB, BB and hybrid models - Case L3 (Kotoulas et al.).

the constant prior. However, the combination of a WB model, even if its predictions are not very accurate for the case under study, with a GP, can be highly beneficial. More precisely, as new data are generated, the GP component of the hybrid model will consider them and correct



(a) 160°C

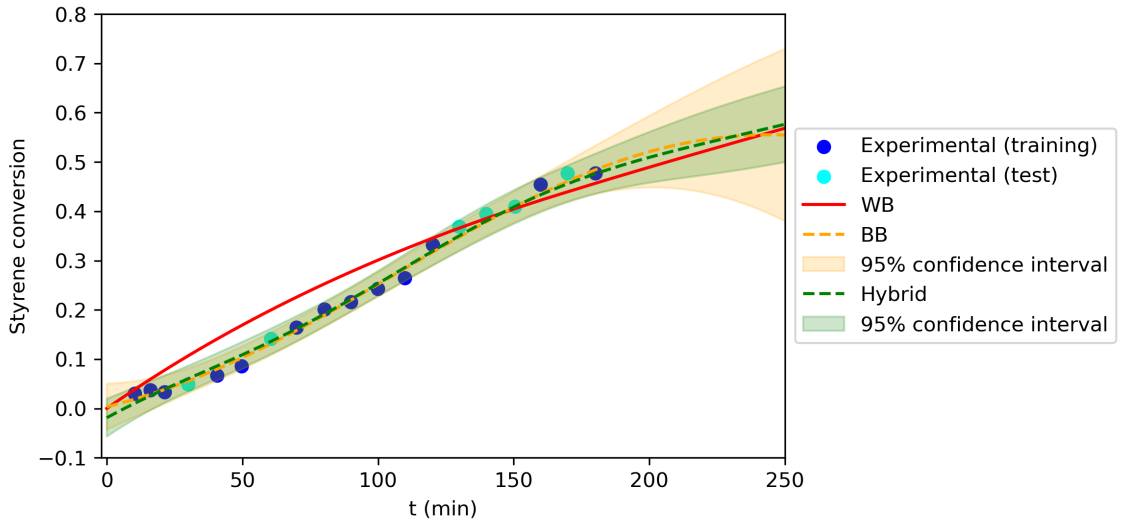
(b) 170°C

(c) 180°C

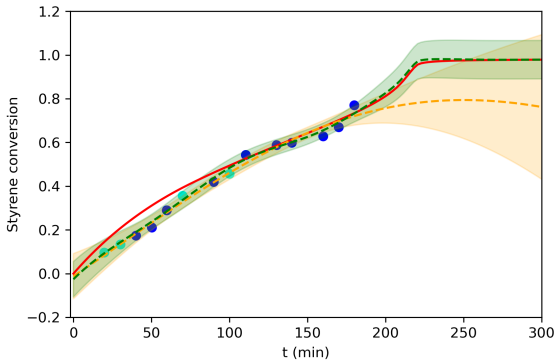
Figure 5.8: Comparison of WB, BB and hybrid models - Case L4 (Kotoulas et al.).

the model predictions along the running process. What's more, the confidence estimations will adapt to the new data not being consistent with the prior (i.e., the predictions of the WB model). This update can be performed in a continuous manner, under the light of new incoming data, and steadily improve the overall model accuracy in a very efficient and rapid manner. An example of such a case, in which the training data are not scattered along the domain but are introduced progressively and in the order of their appearance to the three models is presented in Figure 5.12 for L4 case at 170°C. As can be seen, despite the failure of the WB model to simulate with accuracy the system behavior, the GP model corrects this behavior and compensates its weakness. On the other hand, the BB model, as a standalone, is not very useful, especially in the early stages of the process, as data are very few and the model converges directly to the prior. Note that, the points appearing as "test" points here are not used at all during the model development or testing; they are just shown here to illustrate the "future" (i.e., unknown to the model) evolution of the system.

Finally, to have a more global evaluation of the performance of the three models, 100 different training/test splits are performed for literature cases. The average and the standard deviation of the test MAE obtained for the different models are shown in Table 5.5. These results reflect



(a) 80°C

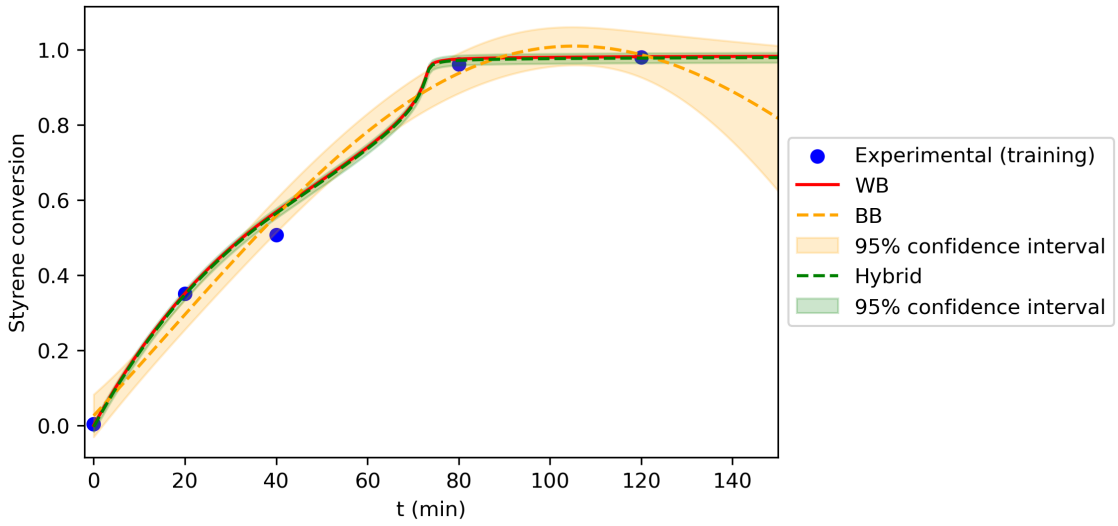


(b) 90°C

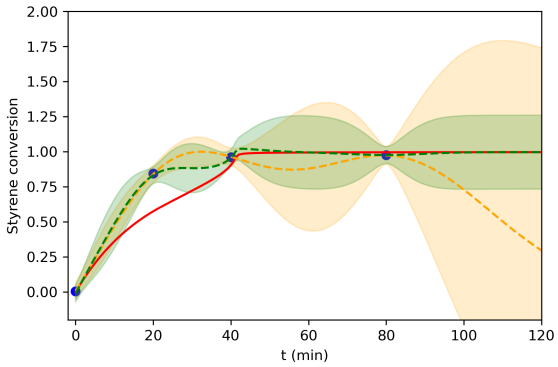
Figure 5.9: Comparison of WB, BB and hybrid models - Case L5 (Vicevic et al.).

both the performance within and outside the domain, as among the different random splits, some have test data in extrapolation in comparison to the training data. The best prediction accuracy is obtained for the hybrid model which surpasses pure BB and WB models in most cases, and provides similar performances in few other cases. Consequently, the hybrid structure offers great potential for the modeling of the styrene-GTR graft polymerization, combining the advantages of both the BB and WB model. However, the main drawback of the hybrid model is that its performance is highly dependent on how each of its components is built and how the connection is made between them.

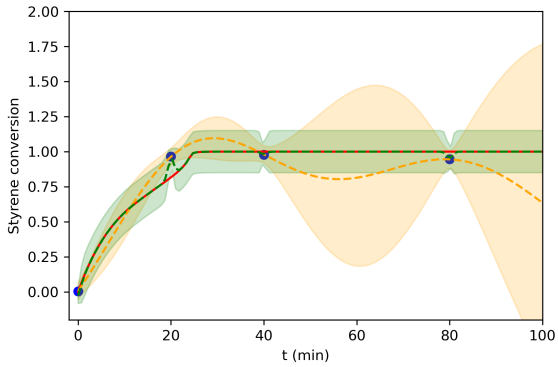
The presented results are preliminary and further research should be pursued to improve and better understand the procedure. Several ways of improvement are proposed in the following.



(a) Case E1, 90°C



(b) Case E2, 100°C

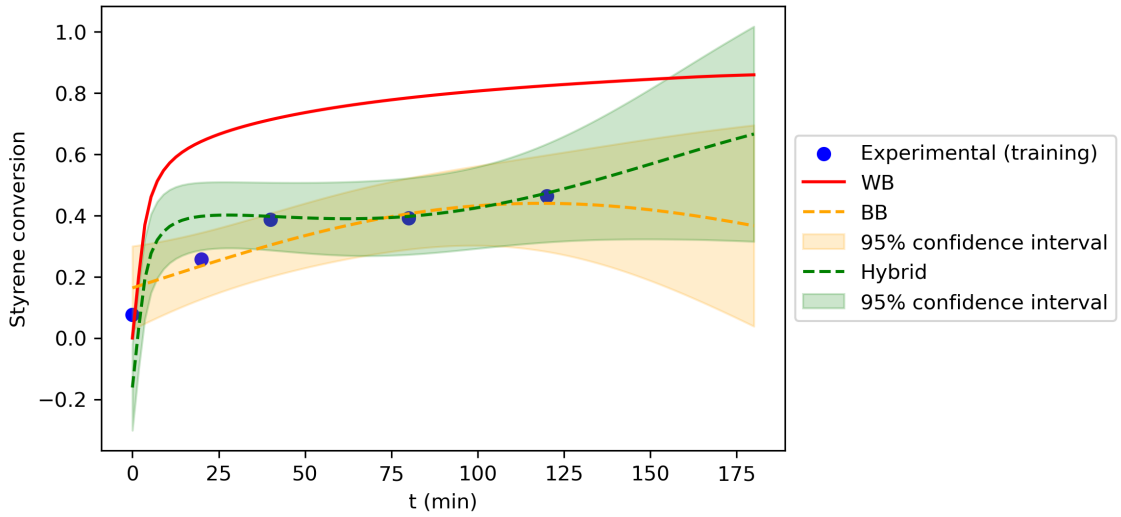


(c) Case E3, 110°C

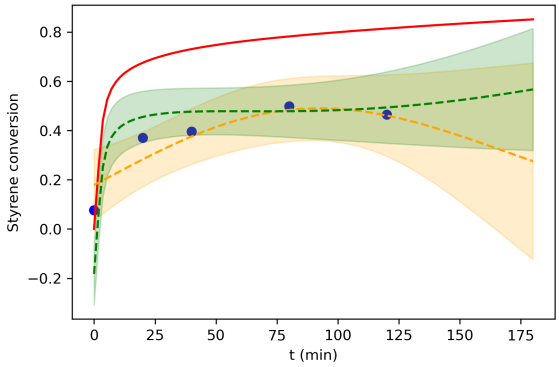
Figure 5.10: Comparison of WB, BB and hybrid models - Cases E1/E2/E3.

5.5 Conclusions and ways of improvement

In this preliminary work, a hybrid model was constructed for the styrene-GTR graft polymerization system. To do so, a black-box (BB) Gaussian Processes (GP) model was used to predict the mismatch between the experimental measurements and the predictions provided by a white-box (WB) model. The latter contained material balances, the detailed kinetics of the reactions and the gel-/cage-/glass- effects occurring in the polymerization process. For the majority of the different investigated data sets, the hybrid structure displayed better performances, both in interpolation and extrapolation, than the pure WB or BB models. Indeed, on the one hand, the system's behavior is not fully understood and the WB might therefore contain a poor description of the mechanisms (especially due to GTR complex structure and composition) and/or the important phenomena and interactions. The experimental errors, that are inherent in the used data, also impart a level of uncertainty in the prediction of the model parameters. These could cause deviations, between the WB model predictions and the observations, when being applied to a slightly different system. On the other hand, a pure BB, besides being prone to the same issues related to data uncertainty, can lead to physically inconsistent predictions. The



(a) Case E4, 90°C



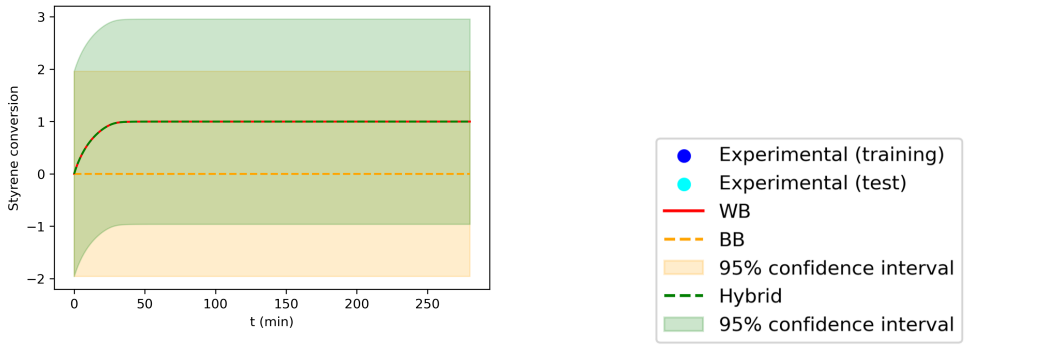
(b) Case E5, 100°C

Figure 5.11: Comparison of WB, BB and hybrid models - Cases E4/E5.

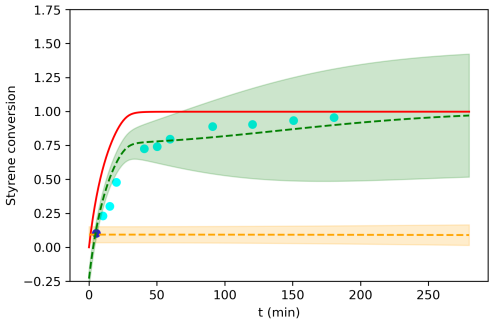
hybrid modeling enables to combine the advantages of both the WB and BB models such as interpretability, extrapolation capability, computation time, model complexity and data requirement. At the same time, this study highlights the importance of the accuracy of the WB model in the adopted hybrid structure.

To improve and better understand the hybrid modeling procedure, several ways of improvement are proposed in the following:

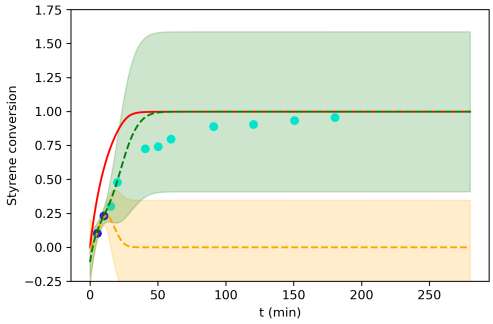
- **Prior mean and covariance functions:** Basic prior mean and covariance functions were utilized in this work. However, several methods are proposed in the literature to customize this prior distribution and could be investigated. Some examples are: a more straightforward implementation of the WB model as the prior mean, use of mean and covariance functions learned from data and physical constraints, use of composite kernels.
- **GP module/library** The GP model was exclusively based on the one available in Scikit-Learn library. However, other modules/libraries such as *gpytorch* or *GPflow* could be employed, for example to customize the prior mean and covariance functions (not feasible



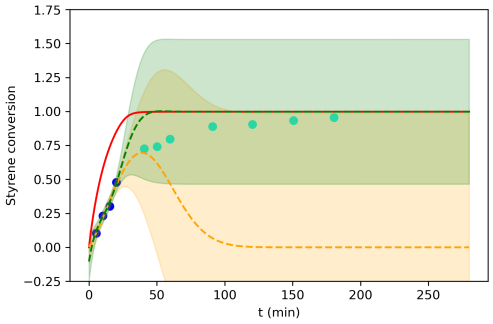
(a) Before the generation of any data



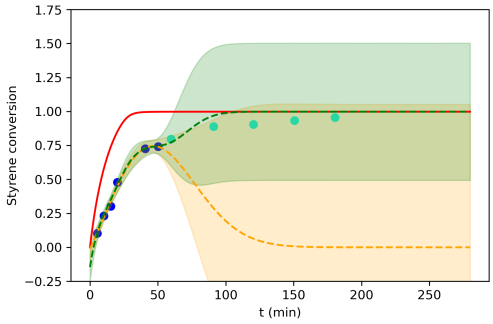
(b) 1 data



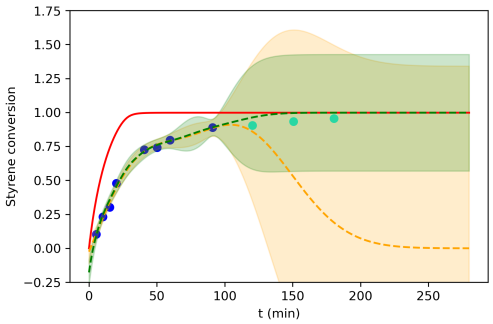
(c) 2 data



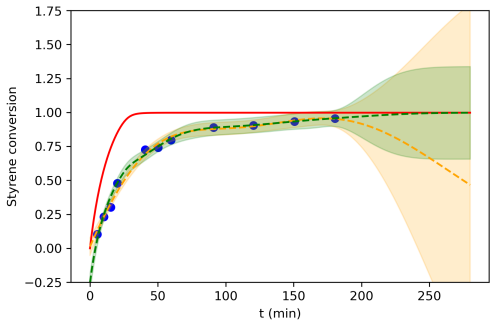
(d) 4 data



(e) 6 data



(f) 8 data



(g) All data

Figure 5.12: Comparison of WB, BB and hybrid models trained on data generated by a running process - Case L4, 170°C. The test points are only for illustration of the "future" evolution of the system (unknown to the model at all stages).

Table 5.5: Comparaison of the test MAE of the WB, BB and hybrid models (averaged over 100 different train/test splits with a ratio 70%/30%).

Case	Subcase	MAE test (%)		
		WB	**BB**	**Hybrid**
L1. Florez et al.	1	4.1 ± 1.1	3.6 ± 1.5	2.1 ± 1.0
	2	3.6 ± 0.9	2.7 ± 0.9	1.8 ± 0.8
	3	1.9 ± 1.2	5.0 ± 5.4	2.0 ± 1.3
L2. Villalobos et al.	-	1.1 ± 0.5	2.6 ± 2.0	0.9 ± 0.4
L3. Kotoulas et al.	1	4.8 ± 2.2	4.1 ± 3.0	2.2 ± 1.9
	2	3.0 ± 1.5	3.9 ± 4.6	2.3 ± 1.5
	3	3.5 ± 1.3	5.6 ± 4.9	1.9 ± 0.9
	4	2.7 ± 1.0	4.2 ± 4.1	2.3 ± 0.8
L4. Kotoulas et al.	1	23.7 ± 4.5	5.7 ± 6.2	5.4 ± 2.2
	2	21.6 ± 6.3	7.7 ± 5.7	5.4 ± 1.9
	3	26.1 ± 6.4	3.9 ± 4.9	4.7 ± 2.4
L5. Vicevic et al.	1	4.1 ± 0.9	1.5 ± 0.3	1.6 ± 0.4
	2	4.6 ± 1.2	2.7 ± 1.1	2.8 ± 0.8

with Scikit-Learn).

- **Constrained GP:** Some literature reports demonstrate the positive effect of adding constraints (e.g., monotonicity, inequalities, boundaries) on the GP parameters to further enhance the performance/physical reliability of the hybrid models. The mathematical implementation is more complex than simple GP models but some modules were developed such as *gp_constr*, *gp_tools* or *lineqGPR*. The integration to the hybrid model of the well-known [0-1] boundaries and increasing behavior of reaction conversion will probably provide more physically consistent predictions.

- **Consideration of experimental data uncertainties:** The experimental measurements were accompanied by uncertainties in both the inputs (reactant contents, reaction time) and outputs (reaction conversion). Further research must be implemented to better integrate this uncertainties into the modeling scheme.

- **Hybrid model structure:** In this work, only one hybrid structure was tested. It led to poor performances in the cases where the WB model was not precise enough. Therefore, other hybrid configurations could be considered. For example, a serial configuration BB/WB could be envisioned and would require a re-development of the WB model that includes parts of the WB model predicted by the BB model (e.g., reactions that were removed in the simplified kinetic scheme). As for the WB/BB serial configuration, the possible implementations are the use of prior mean and covariance functions based or learned from the WB model, but also the adding of knowledge-based constraints to the Gaussian Processes model. Besides, the effects of incorporating different levels of knowledge can be studied for the different hybrid configurations (e.g., elimination of the not-well understood parts in the WB model). This will indeed provide insights on the most adapted hybrid configurations given the precision of the WB model.

- **Industrial application/Evolving GP:** The hybrid modeling procedure could be investigated for industrial application, in a situation where data is collected one-by-one along

the production phase, for example to control the quality of a product. In this sense, an application could be to check whether the hybrid model can well adapt to the incoming new data and eventually predict the output at a remote time with only few data at initial times (e.g., Figure 5.12).

- **Application to other properties:** Additional applications for the prediction of the grafting efficiency of PS on GTR and of the PS average molecular weight could be envisioned with the developed hybrid model.

Abbreviations

ANN	Artificial neural networks
BB	Black-box
BPO	Benzoyl peroxide
DCP	Dicumyl peroxide
DT	Decision tree
GP	Gaussian processes
GTR	Ground tire rubber
MAE	Mean absolute error
MI	Melt index
ML	Machine learning
OECD	Organisation for Economic Co-operation and Development
QSAR	Quantitative structure-activity relationship
RBF	Radial basis function
SVM	Support vector machines
PLS	Partial least squares
PS	Polystyrene
WB	White-box

Appendix A. Additional experimental data

Table A1: Additional experimental data in absence of GTR (cases E1/E2/E3).

N°	t (min)	T (°C)	miGTR EXP (g)	miBPO EXP (g)	miSty EXP (g)	Styrene conversion
1	0	-	0	0.079	1.424	0.0090
2	0	-	0	0.079	1.428	0.0000
3	0	-	0	0.078	1.421	0.0028
4	20	90	0	0.079	1.435	0.3345
5	20	90	0	0.079	1.443	0.3792
6	20	90	0	0.079	1.510	0.3398
7	40	90	0	0.079	1.412	0.5286
8	40	90	0	0.079	1.409	0.5092
9	40	90	0	0.079	1.409	0.4837
10	80	90	0	0.078	1.358	0.9396
11	80	90	0	0.078	1.263	0.9774
12	80	90	0	0.079	1.359	0.9686
13	120	90	0	0.078	1.302	0.9825
14	120	90	0	0.079	1.346	0.9738
15	120	90	0	0.079	1.326	0.9827
16	20	100	0	0.078	1.405	0.8986
17	20	100	0	0.079	1.437	0.7634
18	20	100	0	0.078	1.427	0.8696
19	40	100	0	0.078	1.451	0.9418
20	40	100	0	0.078	1.387	0.9791
21	40	100	0	0.079	1.435	0.9654
22	80	100	0	0.079	1.367	0.9747
23	20	110	0	0.078	1.387	0.9663
24	40	110	0	0.078	1.360	0.9793
25	80	110	0	0.079	1.661	0.9462

Table A2: Additional experimental data in presence of GTR (cases E4/E5).

N°	**t (min)**	**T (°C)**	**miGTR EXP (g)**	**miBPO EXP (g)**	**miSty EXP (g)**	**Styrene conversion**
1	0	-	0.143	0.071	1.290	0.0788
2	0	-	0.143	0.070	1.283	0.0854
3	0	-	0.143	0.071	1.290	0.0861
4	0	-	0.144	0.071	1.286	0.0540
5	0	-	0.144	0.071	1.286	0.0552
6	0	-	0.145	0.070	1.286	0.1175
7	0	-	0.144	0.070	1.286	0.1196
8	0	-	0.143	0.071	1.292	0.0135
9	20	90	0.143	0.071	1.283	0.2559
10	20	90	0.143	0.071	1.300	0.2499
11	20	90	0.143	0.071	1.323	0.2661
12	40	90	0.142	0.072	1.221	0.3839
13	40	90	0.143	0.071	1.302	0.3844
14	40	90	0.143	0.072	1.281	0.3920
15	80	90	0.144	0.071	1.151	0.3909
16	120	90	0.145	0.071	1.099	0.4644
17	120	90	0.142	0.071	1.310	0.4690
18	120	90	0.142	0.071	1.184	0.4578
19	20	100	0.145	0.071	1.288	0.3707
20	40	100	0.144	0.071	1.275	0.3796
21	40	100	0.142	0.071	1.271	0.4022
22	40	100	0.142	0.071	1.282	0.3845
23	40	100	0.144	0.071	1.264	0.3951
24	40	100	0.144	0.071	1.294	0.4208
25	80	100	0.142	0.070	1.149	0.4838
26	80	100	0.142	0.071	1.170	0.4849
27	80	100	0.144	0.071	1.126	0.5269
28	120	100	0.142	0.072	1.326	0.4568
29	120	100	0.143	0.071	1.310	0.4697
30	120	100	0.144	0.071	1.202	0.4664

Appendix B. Detailed WB model

Appendix B1. Polymerization kinetic mechanism [58]

- Chemical initiation
  - Thermal decomposition of initiator:

$$I \xrightarrow{k_d, f_1} 2PR \tag{B1}$$

  - Decomposition of initiator induced by carbon black:

$$I + CB \xrightarrow{k_{dCB}, f_2} PRl \tag{B2}$$

  - Free radical initiation:

$$M + PR \xrightarrow{k_I} R_1 \tag{B3}$$

  - Formation of grafted primary radicals:

$$G + PR \xrightarrow{k_{IG}} GPR \tag{B4}$$

  - Grafted radical initiation:

$$M + GPR \xrightarrow{k_{pG}} GR_1 \tag{B5}$$

  - Initiation of reduced-activity free radicals:

$$M + PRl \xrightarrow{\alpha \cdot k_I} P_1 \tag{B6}$$

  - Initiation of reduced-activity grafted radicals:

$$G + PRl \xrightarrow{\alpha \cdot k_{IG}} GP_1 \tag{B7}$$

- Thermal initiation
  - Diels-Alder dimerization of styrene:

$$2M \underset{k_{-1}}{\overset{k_1}{\leftrightarrow}} AH \tag{B8}$$

  - Initiation from Diels-Alder adduct (AH):

$$AH + M \xrightarrow{k_2} AR + MR \tag{B9}$$

  - Initiation from 1-phenyltetralyl radical (AR):

$$AR + M \xrightarrow{k_A} R_3 \tag{B10}$$

  - Initiation from styryl radical (MR):

$$MR + M \xrightarrow{k_B} R_2 \tag{B11}$$

  - Trimerization reaction of Diels-Alder adduct:

$$AH + M \xrightarrow{k_C} D_3 \tag{B12}$$

- Propagation:
  - of free radicals:

$$M + R_n \xrightarrow{k_p} R_{n+1} \tag{B13}$$

  - of grafted radicals:

$$M + GR_n \xrightarrow{k_{pG}} GR_{n+1} \tag{B14}$$

  - of reduced-activity free radicals:

$$M + P_n \xrightarrow{\alpha \cdot k_p} P_{n+1} \tag{B15}$$

  - of reduced-activity grafted radicals:

$$M + GP_n \xrightarrow{\alpha \cdot k_{pG}} GP_{n+1} \tag{B16}$$

- Transfer:
  - to monomer from free radicals:

$$M + R_n \xrightarrow{k_{fm}} D_n + R_1 \tag{B17}$$

  - to adduct from free radicals:

$$AH + R_n \xrightarrow{k_{fAH}} D_n + AR \tag{B18}$$

  - to monomer from grafted radicals:

$$M + GR_n \xrightarrow{k_{fmG}} GD_n + R_1 \tag{B19}$$

  - to adduct from grafted radicals:

$$AH + GR_n \xrightarrow{k_{fAHG}} GD_n + AR \tag{B20}$$

  - to GTR from free radicals:

$$G + R_n \xrightarrow{k_{fG}} D_n + GPR \tag{B21}$$

- Scission (induced by primary-radicals):
  - of free polymer

$$PR + D_n \xrightarrow{k_s} R_m + D_{n-m} \tag{B22}$$

  - of grafted polymer

$$PR + GD_n \xrightarrow{k_{sG}} R_m + GD_{n-m} \tag{B23}$$

- Termination:

- by combination of free radicals

$$R_n + R_m \xrightarrow{k_{tc}} D_{n+m} \tag{B24}$$

- by disproportionation of free radicals

$$R_n + R_m \xrightarrow{k_{td}} D_n + D_m \tag{B25}$$

- by combination between free and grafted radicals

$$R_n + GR_m \xrightarrow{k_{tcG}} GD_{n+m} \tag{B26}$$

- by disproportionation between free and grafted radicals

$$R_n + GR_m \xrightarrow{k_{tdG}} D_n + GD_m \tag{B27}$$

- between free and reduced-activity radicals

$$R_n + P_m \xrightarrow{\alpha \cdot k_{tc/td}} D_{n+m}/D_n + D_m \tag{B28}$$

- between grafted and reduced-activity radicals

$$GR_n + P_m \xrightarrow{\alpha \cdot k_{tcG/tdG}} GD_{n+m}/GD_n + D_m \tag{B29}$$

- between free and reduced-activity grafted radicals

$$R_n + GP_m \xrightarrow{\alpha \cdot k_{tcG/tdG}} GD_{n+m}/D_n + GD_m \tag{B30}$$

- between reduced-activity radicals

$$P_n + P_m \xrightarrow{\alpha^2 \cdot k_{tc/td}} D_{n+m}/D_n + D_m \tag{B31}$$

- between free and primary-radicals

$$R_n + PR \xrightarrow{k_{tPR}} D_n \tag{B32}$$

- between grafted and primary-radicals

$$GR_n + PR \xrightarrow{k_{tPRG}} GD_n \tag{B33}$$

- Radical deactivation by carbon black:
  - of primary-radicals

$$PR/GPR + CB \xrightarrow{k_{dea}} DPR \tag{B34}$$

  - of MR radicals

$$MR + CB \xrightarrow{k_{dea}} D_1 \tag{B35}$$

  - of AR radicals

$$AR + CB \xrightarrow{k_{dea}} D_2 \tag{B36}$$

– of free radicals

$$CB + R_n \xrightarrow{k_{dea}} D_n \tag{B37}$$

– of grafted radicals

$$CB + GR_n \xrightarrow{k_{deaG}} GD_n \tag{B38}$$

– of reduced-activity free radicals

$$CB + P_n \xrightarrow{\alpha \cdot k_{dea}} D_n \tag{B39}$$

– of reduced-activity grafted radicals

$$CB + GP_n \xrightarrow{\alpha \cdot k_{deaG}} GD_n \tag{B40}$$

Appendix B2. Model parameters [58]

Table B1: Model parameters

Parameter	Value	Units	Reference
$k_{d,DCP}$	$5.500 \cdot 10^{17} \cdot exp(-36650/R/T)$	min^{-1}	[44]
$k_{d,BPO}$	$2.289 \cdot 10^{14} \cdot exp(-27233/R/T)$	min^{-1}	[89]
k_p	$1.149 \cdot 10^{9} \cdot exp(-7513/R/T)$	$l.mol^{-1}.min^{-1}$	This work [58]
k_{-1}	$6.840 \cdot 10^{3} \cdot exp(-13533/R/T)$	min^{-1}	[44]
k_1/k_{-1}	$6.400 \cdot 10^{4} \cdot exp(-12907/R/T)$	$l.mol^{-1}$	[44]
k_2	$9.800 \cdot 10^{7} \cdot exp(-23883/R/T)$	$l.mol^{-1}.min^{-1}$	[44]
k_C	$2.360 \cdot 10^{6} \cdot exp(-21346/R/T)$	$l.mol^{-1}.min^{-1}$	This work [58]
k_{fm}	$1.032 \cdot 10^{7} \cdot exp(-10545/R/T)$	$l.mol^{-1}.min^{-1}$	This work [58]
k_{fAH}	$7.360 \cdot 10^{7} \cdot exp(-30800/R/T)$	$l.mol^{-1}.min^{-1}$	This work [58]
k_{tc}	$7.530 \cdot 10^{10} \cdot exp(-1677/R/T)$	$l.mol^{-1}.min^{-1}$	[44]
k_{dCB}	$2.147 \cdot 10^{-1} \cdot exp(-1478.7/R/T)$	$l.mol^{-1}.min^{-1}$	This work [58]
$log(k_{pG}/k_p)$	$5.34 \cdot 10^{-2} \cdot T(°C) - 5.1574$	-	This work [58]
$log(k_{IG}/k_p)$	$7.52 \cdot 10^{-2} \cdot T(°C) - 7.1353$	-	This work [58]
k_{tcG}/k_{tc}	9.823	-	This work [58]
k_{dea}/k_p	0.310	-	This work [58]
k_{deaG}/k_{dea}	3.770	-	This work [58]
α	$3.70 \cdot 10^{-6}$	-	This work [58]
f_1	0.65 (0.83 for DSC data)	-	This work [58]
f_2	0.40	-	This work [58]
f_{GTR}	0.53	-	This work [58]
$GRat_c$	0.43	-	This work [58]
$a_m; a_p$	$1.4 \cdot 10^{-3}; 4.8 \cdot 10^{-4}$	K^{-1}; K^{-1}	[92], this work [58]
$\delta; \sigma$	$3.8 \cdot 10^{-9}; 3.7 \cdot 10^{-9}$	$m; m$	This work [58]
$j_c; \delta_c$	$175; 5.0 \cdot 10^{-4}$	$-; l.g^{-1}$	[92], this work [58]
$A_{cr}; E_{cr}/R$	9.0; 1960	$g^{1/2}.mol^{-1/2}; K$	This work [58]
$A_{crm}; E_{crm}$	0.231; 1670	$-; cal.mol^{-1}$	[92], this work [58]
$A; B; C/B; n$	0.4315; 0.7; 0.9; 2.5	$-; -; -; -$	This work [58]

$R = 1.9872\ cal.mol^{-1}.K^{-1}$

General conclusions and perspectives

The positive contribution of ML methods to the solving of many problems in Chemical Product Engineering (CPE) is obvious. Indeed, many works in the literature and those carried out within the framework of this thesis have given promising results on problems where classical approaches are not applicable. However, numerous challenges were also identified, through the state-of-the-art and the two applications studied. These challenges are mainly characteristic of the CPE field.

Firstly, the **representation of chemical data** (e.g., molecules, reactions, spectra, sensory properties) is an active area of research. In the first case study in this thesis, molecules were represented by descriptors and the models built performed well compared to other representations and models developed in the literature. Nevertheless, each type of representation has its own advantages and disadvantages, and their choice will depend on each problem. One potential improvement for the first case study therefore lies in the molecular representation (improving the descriptor-based representation or testing other types of representations such as graph neural networks).

Another important area of research concerns the **development of hybrid models** that combine ML approaches with existing knowledge-based approaches. This type of models is interesting for the CPE domain, where a partial knowledge of the systems and only few data are often available, as it overcomes the major shortcomings of ML (i.e., lack of interpretability, poor extrapolability, large amount of data required) while avoiding the complex development of knowledge-based models. However, the way in which knowledge is integrated remains an area for improvement, as in the second application of this thesis, where only one hybrid configuration was tested. A hybrid approach could also be of interest to improve the interpretability and extrapolability of the QSPR models developed in the first application. More generally, the trend is towards finding methods to improve the interpretability of ML models, as shown by several recent studies [23, 25, 59, 61].

Data availability is another critical issue in some CPE applications, for example when data is generated experimentally, when a system has been poorly investigated or due to confidentiality/competitiveness issues. For this reason, a number of ML approaches, more suitable to small data sets, are being investigated such as transfer learning, active learning, semi-supervised learning or hybrid modeling. In addition, the sharing of scientific data is increasingly encouraged within the scientific community. These data should include positive (and negative) results, as well as complete information on the method used to generate the data and its uncertainties. Better data availability would also render benchmarking easier.

The following elements also deserve further study. As in the two applications of this thesis, CPE data are often accompanied by **uncertainties** that would be interesting to be able to properly integrate into ML models to increase their ability to generalize. In the same way, accompanying ML model predictions with their uncertainties would enable to obtain predictions that are more representative of the physical reality. In this sense, the Gaussian Processes studied in the second application represent a possible option.

An often overlooked but important aspect is the **applicability domain** of the models, as demonstrated in the first case study. In the context of QSPR models, this aspect is gaining greater importance to improve their utilization for example in the regulation of chemicals. However, this aspect is often overlooked and more appropriate approaches are needed, especially in high dimension.

More generally, ML possesses its own limitations and requirements, which one should be aware of before starting any ML project. Moreover, the development of a ML model is not just a matter of training a model on data and getting it to perform well. On the contrary, a ML project is made up of several stages, which are all very important. Data collection and preparation to have data in sufficient quality and quantity is a first challenge. Then, many possible methods exist at each stage of model development, depending on the characteristics of the data, and each choice has an impact on the performance of the final model. However, there is no precise guide in going about these choices; the case studies presented in this thesis show some of the possible approaches that can be employed in this sense.

Finally, it is certain that future research will progressively provide a better understanding of the use of ML approaches, and exploit the full potential of these methods to find solutions adapted to the complex challenges of CPE.

Bibliography

[1] https://medium.com/@roiyeho/random-forests-98892261dc49. Accessed: 2023-10-01.

[2] https://towardsdatascience.com/how-to-perform-anomaly-detection-with-the-isolation-f Accessed: 2023-10-01.

[3] https://www.ustires.org/whats-tire-0. Accessed: 2023-10-01.

[4] https://scipy-lectures.org/packages/scikit-learn/auto_examples/plot_tsne.html. Accessed: 2023-10-01.

[5] C. Agrell. Gaussian processes with linear operator inequality constraints. *Journal of Machine Learning Research*, 20, 1–36, 2019.

[6] A. Al Mohtar, M. L. Pinto, A. Neves, S. Nunes, D. Zappi, G. Varani, A. M. Ramos, M. J. Melo, N. Wallaszkovits, J. I. Lahoz Rodrigo, K. Herlt, et J. Lopes. Decision making based on hybrid modeling approach applied to cellulose acetate based historical films conservation. *Scientific Reports*, 11(1), 1–13, 2021, http://dx.doi.org/10.1038/s41598-021-95373-0.

[7] G. Arzamendi, A. D'Anjou, M. Graña, J. R. Leiza, et J. M. Asua. Model reduction in emulsion polymerization using hybrid first principles/artificial neural networks models, 2 long chain branching kinetics. *Macromolecular Theory and Simulations*, 14(2), 125–132, 2005, http://dx.doi.org/10.1002/mats.200400064.

[8] D. J. Audus, A. McDannald, et B. Decost. Leveraging Theory for Enhanced Machine Learning. *ACS Macro Letters*, 11(9), 1117–1122, 2022, http://dx.doi.org/10.1021/acsmacrolett.2c00369.

[9] R. Batra, H. Dai, T. D. Huan, L. Chen, C. Kim, W. R. Gutekunst, L. Song, et R. Ramprasad. Polymers for Extreme Conditions Designed Using Syntax-Directed Variational Autoencoders. *Chemistry of Materials*, 32(24), 10489–10500, 2020, http://dx.doi.org/10.1021/acs.chemmater.0c03332.

[10] X. Cao et R. Yousefzadeh. Extrapolation and AI transparency: Why machine learning models should reveal when they make decisions beyond their training. *Big Data and Society*, 10(1), 2023, http://dx.doi.org/10.1177/20539517231169731.

[11] L. L. T. Chan et J. Chen. Melt index prediction with a mixture of Gaussian process regression with embedded clustering and variable selections. *Journal of Applied Polymer Science*, 134(40), 1–11, 2017, http://dx.doi.org/10.1002/app.45237.

[12] C. Chang et T. Zeng. A hybrid data-driven-physics-constrained Gaussian process regression framework with deep kernel for uncertainty quantification. *Journal of Computational Physics*, 486, 112129, 2023, http://dx.doi.org/10.1016/j.jcp.2023.112129.

[13] L. Chen, C. Kim, R. Batra, J. P. Lightstone, C. Wu, Z. Li, A. A. Deshmukh, Y. Wang, H. D. Tran, P. Vashishta, G. A. Sotzing, Y. Cao, et R. Ramprasad. Frequency-dependent dielectric constant prediction of polymers using machine learning. *npj Computational Materials*, 6(1), 30–32, 2020, http://dx.doi.org/10.1038/s41524-020-0333-6.

[14] T. Chen et J. Ren. Bagging for Gaussian process regression. *Neurocomputing*, 72(7-9), 1605–1610, 2009, http://dx.doi.org/10.1016/j.neucom.2008.09.002.

[15] M. Civera, G. Boscato, et L. Zanotti Fragonara. Treed gaussian process for manufacturing imperfection identification of pultruded GFRP thin-walled profile. *Composite Structures*, 254(August), 112882, 2020, http://dx.doi.org/10.1016/j.compstruct.2020.112882.

[16] J. Dai et R. V. Krems. Interpolation and Extrapolation of Global Potential Energy Surfaces for Polyatomic Systems by Gaussian Processes with Composite Kernels. *Journal of Chemical Theory and Computation*, 16(3), 1386–1395, 2020, http://dx.doi.org/10.1021/acs.jctc.9b00700.

[17] W. Dai, S. Mohammadi, et S. Cremaschi. A hybrid modeling framework using dimensional analysis for erosion predictions. *Computers and Chemical Engineering*, 156, 107577, 2022, http://dx.doi.org/10.1016/j.compchemeng.2021.107577.

[18] H. Deng, Y. Liu, P. Li, et S. Zhang. Hybrid model for discharge flow rate prediction of reciprocating multiphase pumps. *Advances in Engineering Software*, 124(August), 53–65, 2018, http://dx.doi.org/10.1016/j.advengsoft.2018.08.006.

[19] Y. Deng, C. Avila, H. Gao, I. Mantilla, M. R. Eden, et S. Cremaschi. A hybrid modeling approach to estimate liquid entrainment fraction and its uncertainty. *Computers and Chemical Engineering*, 162, 107796, 2022, http://dx.doi.org/10.1016/j.compchemeng.2022.107796.

[20] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti, et G. Csányi. Gaussian Process Regression for Materials and Molecules. *Chemical Reviews*, 121(16), 10073–10141, 2021, http://dx.doi.org/10.1021/acs.chemrev.1c00022.

[21] L. Ding, S. Mak, et C. F. J. Wu. BdryGP: a new Gaussian process model for incorporating boundary information. 2019.

[22] H. Doan Tran, C. Kim, L. Chen, A. Chandrasekaran, R. Batra, S. Venkatram, D. Kamal, J. P. Lightstone, R. Gurnani, P. Shetty, M. Ramprasad, J. Laws, M. Shelton, et R. Ramprasad. Machine-learning predictions of polymer properties with Polymer Genome. *Journal of Applied Physics*, 128(17), 2020, http://dx.doi.org/10.1063/5.0023759.

[23] F. Doshi-Velez et B. Kim. Towards A Rigorous Science of Interpretable Machine Learning. (Ml), 1–13, 2017.

[24] F. J. Doyle, C. A. Harrison, et T. J. Crowley. Hybrid model-based approach to batch-to-batch control of particle size distribution in emulsion polymerization. 27, 1153–1163, 2003, http://dx.doi.org/10.1016/S0098-1354(03)00043-7.

[25] M. Du, N. Liu, et X. Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1), 68–77, 2020, http://dx.doi.org/10.1145/3359786.

[26] D. K. Duvenaud. Automatic Model Construction with Gaussian Processes. *PhD Thesis, University of Cambridge*, (June), XIII, 144, 2014.

[27] O. Erge et E. van Oort. Combining physics-based and data-driven modeling in well construction: Hybrid fluid dynamics modeling. *Journal of Natural Gas Science and Engineering*, 97(June 2021), 104348, 2022, http://dx.doi.org/10.1016/j.jngse.2021.104348.

[28] D. Florez, S. Hoppe, G. H. Hu, et D. Meimaroglou. Radical bulk polymerization of styrene in the presence of rubber particles from recycled tires: a kinetic study using DSC. *Journal of Thermal Analysis and Calorimetry*, 2020, http://dx.doi.org/10.1007/s10973-020-09701-z.

[29] D. C. Florez Parra. Effects of the presence of recycled tire powders on the kinetics of the radical polymerization of styrene and the properties of the resulting materials. 1–197, 2019.

[30] C. Furtado, L. F. Pereira, R. P. Tavares, M. Salgado, F. Otero, G. Catalanotti, A. Arteiro, M. A. Bessa, et P. P. Camanho. A methodology to generate design allowables of composite laminates using machine learning. *International Journal of Solids and Structures*, 233 (October 2020), 111095, 2021, http://dx.doi.org/10.1016/j.ijsolstr.2021.111095.

[31] Z. Ge, T. Chen, et Z. Song. Quality prediction for polypropylene production process based on CLGPR model. *Control Engineering Practice*, 19(5), 423–432, 2011, http://dx.doi.org/10.1016/j.conengprac.2011.01.002.

[32] A. Giuntoli, N. K. Hansoge, A. van Beek, Z. Meng, W. Chen, et S. Keten. Systematic coarse-graining of epoxy resins with machine learning-informed energy renormalization. *npj Computational Materials*, 7(1), 2021, http://dx.doi.org/10.1038/s41524-021-00634-1.

[33] C. Go, Y. J. Kwak, S. Kwag, S. Eem, S. Lee, et B. S. Ju. On developing accurate prediction models for residual tensile strength of GFRP bars under alkaline-concrete environment using a combined ensemble machine learning methods. *Case Studies in Construction Materials*, 18(April), e02157, 2023, http://dx.doi.org/10.1016/j.cscm.2023.e02157.

[34] J. Görtler, R. Kehlbeck, et O. Deussen. A visual exploration of gaussian processes. *Distill*, 2019, http://dx.doi.org/10.23915/distill.00017.

[35] M. Hinchliffe, G. Montague, M. Willis, et A. Burke. Hybrid Approach to Modeling an Industrial Polyethylene Process. *AIChE Journal*, 49(12), 3127–3137, 2003, http://dx.doi.org/10.1002/aic.690491213.

[36] S. J. Hong, H. Chun, J. Lee, B. H. Kim, M. H. Seo, J. Kang, et B. Han. First-Principles-Based Machine-Learning Molecular Dynamics for Crystalline Polymers with van der Waals Interactions. *Journal of Physical Chemistry Letters*, 12, 6000–6006, 2021, http://dx.doi.org/10.1021/acs.jpclett.1c01140.

[37] A. Hosseini, M. Oshaghi, et S. Engell. Mid-course control of particle size distribution in emulsion polymerization using a hybrid model. *Proceedings of the IEEE International Conference on Control Applications*, 728–733, 2013, http://dx.doi.org/10.1109/CCA.2013.6662836.

[38] A. Jha, A. Chandrasekaran, C. Kim, et R. Ramprasad. Impact of dataset uncertainties on machine learning model predictions: The example of polymer glass transition temperatures. *Modelling and Simulation in Materials Science and Engineering*, 27(2), 24002, 2019, http://dx.doi.org/10.1088/1361-651X/aaf8ca.

[39] C. Jidling, N. Wahlström, A. Wills, et T. B. Schön. Linearly constrained Gaussian processes. *Advances in Neural Information Processing Systems*, 2017-December(Nips), 1216–1225, 2017.

[40] M. Jones et N. Clarke. Machine learning real space microstructure characteristics from scattering data. *Soft Matter*, 17(42), 9689–9696, 2021, http://dx.doi.org/10.1039/d1sm00818h.

[41] C. Kim, A. Chandrasekaran, T. D. Huan, D. Das, et R. Ramprasad. Polymer Genome: A Data-Powered Polymer Informatics Platform for Property Predictions. *Journal of Physical Chemistry C*, 122(31), 17575–17585, 2018, http://dx.doi.org/10.1021/acs.jpcc.8b02913.

[42] J. Kocijan, M. Perne, B. Grašic, M. Z. Božnar, et P. Mlakar. Sparse and hybrid modelling of relative humidity: The Krško basin case study. *CAAI Transactions on Intelligence Technology*, 5(1), 42–48, 2020, http://dx.doi.org/10.1049/trit.2019.0054.

[43] K. Konstadinidis, D. S. Achilias, et C. Kiparissides. Development of a unified mathematical framework for modelling molecular and structural changes in free-radical homopolymerization reactions. *Polymer*, 33(23), 5019–5031, 1992, http://dx.doi.org/10.1016/0032-3861(92)90053-Y.

[44] C. Kotoulas, A. Krallis, P. Pladis, et C. Kiparissides. A comprehensive kinetic model for the combined chemical and thermal polymerization of styrene up to high conversions. *Macromolecular Chemistry and Physics*, 204(10), 1305–1314, 2003, http://dx.doi.org/10.1002/macp.200390104.

[45] T. Krivec, N. Hvala, et J. Kocijan. Integrated theoretical and data-driven Gaussian Process NARX Model for the Simulation of Effluent Concentrations in Wastewater Treatment Plant. *IFAC-PapersOnLine*, 54(7), 714–719, 2021, http://dx.doi.org/10.1016/j.ifacol.2021.08.445.

[46] M. Lange-Hegermann. Linearly Constrained Gaussian Processes with Boundary Conditions. *Proceedings of Machine Learning Research*, 130, 1090–1098, 2021.

[47] Z. Li, X. Hong, K. Hao, L. Chen, et B. Huang. Gaussian process regression with heteroscedastic noises — A machine-learning predictive variance approach. *Chemical Engineering Research and Design*, 157(1996), 162–173, 2020, http://dx.doi.org/10.1016/j.cherd.2020.02.033.

[48] J. P. Lightstone, L. Chen, C. Kim, R. Batra, et R. Ramprasad. Refractive index prediction models for polymers using machine learning. *Journal of Applied Physics*, 127(21), 2020, http://dx.doi.org/10.1063/5.0008026.

[49] Y. Liu et Z. Gao. Industrial melt index prediction with the ensemble anti-outlier just-in-time Gaussian process regression modeling method. *Journal of Applied Polymer Science*, 132(22), 1–10, 2015, http://dx.doi.org/10.1002/app.41958.

[50] Y. Liu, Y. Liang, et Z. Gao. Industrial polyethylene melt index prediction using ensemble manifold learning–based local model. *Journal of Applied Polymer Science*, 134(29), 1–7, 2017, http://dx.doi.org/10.1002/app.45094.

[51] Y. Liu, Z. Xia, H. Deng, et S. Zheng. Two-Stage Hybrid Model for Efficiency Prediction of Centrifugal Pump. *Sensors*, 22(11), 2022, http://dx.doi.org/10.3390/s22114300.

[52] D. Long, Z. Wang, A. Krishnapriyan, R. Kirby, S. Zhe, et M. Mahoney. AutoIP: A United Framework to Integrate Physics into Gaussian Processes. 2022.

[53] A. F. Lopez-Lopera, F. Bachoc, N. Durrande, et O. Roustant. Finite-dimensional Gaussian approximation with linear inequality constraints. *SIAM-ASA Journal on Uncertainty Quantification*, 6(3), 1224–1255, 2018, http://dx.doi.org/10.1137/17M1153157.

[54] C. Lv, J. Huang, M. Zhang, H. Wang, et T. Zhang. Semi-Supervised Deep Kernel Active Learning for Material Removal Rate Prediction in Chemical Mechanical Planarization. *Sensors*, 23(9), 1–21, 2023, http://dx.doi.org/10.3390/s23094392.

[55] L. Marino et A. Cicirello. A switching Gaussian process latent force model for the identification of mechanical systems with a discontinuous nonlinearity. 2023, http://dx.doi.org/10.1017/dce.2023.12.

[56] F. L. Marten et A. E. Hamielec. High-Conversion Diffusion-Controlled Polymerization of Styrene. I. *Journal of Applied Polymer Science*, 27(1), 489–505, 1982.

[57] F. Massa Gray et M. Schmidt. A hybrid approach to thermal building modelling using a combination of Gaussian processes and grey-box models. *Energy and Buildings*, 165, 56–63, 2018, http://dx.doi.org/10.1016/j.enbuild.2018.01.039.

[58] D. Meimaroglou, D. Florez, et G. H. Hu. A kinetic modeling framework for the peroxide-initiated radical polymerization of styrene in the presence of rubber particles from recycled tires. *Chemical Engineering Science*, 248, 117137, 2022, http://dx.doi.org/10.1016/j.ces.2021.117137.

[59] C. Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[60] H. Moriwaki, Y. S. Tian, N. Kawashita, et T. Takagi. Mordred: A molecular descriptor calculator. *Journal of Cheminformatics*, 10(1), 1–14, 2018, http://dx.doi.org/10.1186/s13321-018-0258-y.

[61] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, et B. Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences of the United States of America*, 116(44), 22071–22080, 2019, http://dx.doi.org/10.1073/pnas.1900654116.

[62] V. Nguyen et J. A. Marvel. Modeling of Industrial Robot Kinematics Using a Hybrid Analytical and Statistical Approach. *Journal of Mechanisms and Robotics*, 14(5), 2022, http://dx.doi.org/10.1115/1.4053734.

[63] X. Ou et E. Martin. Batch process modelling with mixtures of Gaussian processes. *Neural Computing and Applications*, 17(5-6), 471–479, 2008, http://dx.doi.org/10.1007/s00521-007-0144-4.

[64] J. Paixão, S. da Silva, E. Figueiredo, L. Radu, et G. Park. Delamination area quantification in composite structures using Gaussian process regression and auto-regressive models. *JVC/Journal of Vibration and Control*, 27(23-24), 2778–2792, 2021, http://dx.doi.org/10.1177/1077546320966183.

[65] A. Patra, R. Batra, A. Chandrasekaran, C. Kim, T. D. Huan, et R. Ramprasad. A multi-fidelity information-fusion approach to machine learn and predict polymer bandgap. *Computational Materials Science*, 172(June 2019), 109286, 2020, http://dx.doi.org/10.1016/j.commatsci.2019.109286.

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

[67] G. Perrin et S. Da Veiga. Constrained Gaussian Process Regression: an Adaptive Approach for the Estimation of Hyperparameters and the Verification of Constraints With High Probability. *Journal of Machine Learning for Modeling and Computing*, 2(2), 55–76, 2021, http://dx.doi.org/10.1615/jmachlearnmodelcomput.2021039837.

[68] S. Pfingstl et M. Zimmermann. On integrating prior knowledge into Gaussian processes for prognostic health monitoring. *Mechanical Systems and Signal Processing*, 171(February), 108917, 2022, http://dx.doi.org/10.1016/j.ymssp.2022.108917.

[69] D. J. Pitchforth, T. J. Rogers, U. T. Tygesen, et E. J. Cross. Grey-box models for wave loading prediction. *Mechanical Systems and Signal Processing*, 159, 107741, 2021, http://dx.doi.org/10.1016/j.ymssp.2021.107741.

[70] R. Planas, N. Oune, et R. Bostanabad. Evolutionary Gaussian Processes. *Journal of Mechanical Design, Transactions of the ASME*, 143(11), 1–12, 2021, http://dx.doi.org/10.1115/1.4050746.

[71] M. Ramprasad et C. Kim. Assessing and Improving Machine Learning Model Predictions of Polymer Glass Transition Temperatures. 3(March), 1–5, 2019.

[72] R. Ramprasad, R. Batra, A. Mannodi-Kanakkithodi, C. Kim, et G. Pilania. Machine learning in materials informatics: Recent applications and prospects. *npj Computational Materials*, 3(1), 2017, http://dx.doi.org/10.1038/s41524-017-0056-5.

[73] C. E. Rasmussen et C. K. I. Williams. *Gaussian Processes for Machine Learning*. 2006, ISBN : 026218253X.

[74] A. W. Rogers, Z. Song, F. V. Ramon, K. Jing, et D. Zhang. Investigating 'greyness' of hybrid model for bioprocess predictive modelling. *Biochemical Engineering Journal*, 190 (November 2022), 108761, 2023, http://dx.doi.org/10.1016/j.bej.2022.108761.

[75] B. Sanchez-Lengeling, L. M. Roch, J. D. Perea, S. Langner, C. J. Brabec, et A. Aspuru-Guzik. A Bayesian Approach to Predict Solubility Parameters. *Advanced Theory and Simulations*, 2(1), 1–10, 2019, http://dx.doi.org/10.1002/adts.201800069.

[76] J. Sansana, M. N. Joswiak, I. Castillo, Z. Wang, R. Rendall, L. H. Chiang, et M. S. Reis. Recent trends on hybrid modeling for Industry 4.0. *Computers and Chemical Engineering*, 151, 107365, 2021, http://dx.doi.org/10.1016/j.compchemeng.2021.107365.

[77] E. Schulz, M. Speekenbrink, et A. Krause. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, 85, 1–16, 2018, http://dx.doi.org/10.1016/j.jmp.2018.03.001.

[78] N. Seryo, J. J. Molina, et T. Taniguchi. Select Applications of Bayesian Data Analysis and Machine Learning to Flow Problems. *Nihon Reoroji Gakkaishi*, 49(2), 97–113, 2021, http://dx.doi.org/10.1678/rheology.49.97.

[79] S. Shahpouri, D. Gordon, C. Hayduk, R. Rezaei, C. R. Koch, et M. Shahbakhti. Hybrid emission and combustion modeling of hydrogen fueled engines. *International Journal of Hydrogen Energy*, 48(62), 24037–24053, 2023, http://dx.doi.org/10.1016/j.ijhydene.2023.03.153.

[80] Z. H. Shen, J. J. Wang, J. Y. Jiang, S. X. Huang, Y. H. Lin, C. W. Nan, L. Q. Chen, et Y. Shen. Phase-field modeling and machine learning of electric-thermal-mechanical breakdown of polymer-based dielectrics. *Nature Communications*, 10(1), 1–10, 2019, http://dx.doi.org/10.1038/s41467-019-09874-8.

[81] M. J. Song, S. H. Ju, S. Kim, S. H. Oh, et J. M. Lee. Hybrid modeling approach for polymer melt index prediction. *Journal of Applied Polymer Science*, (April), 2022, http://dx.doi.org/10.1002/app.52987.

[82] L. P. Swiler, M. Gulian, A. L. Frankel, C. Safta, et J. D. Jakeman. a Survey of Constrained Gaussian Process Regression: Approaches and Implementation Challenges. *Journal of Machine Learning for Modeling and Computing*, 1(2), 119–156, 2020, http://dx.doi.org/10.1615/jmachlearnmodelcomput.2020035155.

[83] L. Tao, V. Varshney, et Y. Li. Benchmarking Machine Learning Models for Polymer Informatics: An Example of Glass Transition Temperature. *Journal of Chemical Information and Modeling*, 61(11), 5395–5413, 2021, http://dx.doi.org/10.1021/acs.jcim.1c01031.

[84] Y. Tian, J. Zhang, et J. Morris. Modeling and optimal control of a batch polymerization reactor using a hybrid stacked recurrent neural network model. *Industrial and Engineering Chemistry Research*, 40(21), 4525–4535, 2001, http://dx.doi.org/10.1021/ie0010565.

[85] H. J. Tulleken. Grey-box modelling and identification using physical knowledge and bayesian techniques. *Automatica*, 29(2), 285–308, 1993, http://dx.doi.org/10.1016/0005-1098(93)90124-C.

[86] F. Valentini et A. Pegoretti. End-of-life options of tyres. A review. *Advanced Industrial and Engineering Polymer Research*, 5(4), 203–213, 2022, http://dx.doi.org/10.1016/j.aiepr.2022.08.006.

[87] S. D. Veiga et A. Marrel. Gaussian process regression with linear inequality constraints. *Reliability Engineering and System Safety*, 195, 2020, http://dx.doi.org/10.1016/j.ress.2019.106732.

[88] M. Vicevic, K. Novakovic, K. V. Boodhoo, et A. J. Morris. Kinetics of styrene free radical polymerisation in the spinning disc reactor. *Chemical Engineering Journal*, 135(1-2), 78–82, 2008, http://dx.doi.org/10.1016/j.cej.2007.05.041.

[89] M. A. Villalobos, A. E. Hamielec, et P. E. Wood. Bulk and suspension polymerization of styrene in the presence of n-pentane. An evaluation of monofunctional and bifunctional initiation. *Journal of Applied Polymer Science*, 50(2), 327–343, 1993, http://dx.doi.org/10.1002/app.1993.070500214.

[90] M. von Stosch, R. Oliveira, J. Peres, et S. Feyo de Azevedo. Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Computers and Chemical Engineering*, 60, 86–101, 2014, http://dx.doi.org/10.1016/j.compchemeng.2013.08.008.

[91] A. G. Wilson et R. P. Adams. Gaussian process kernels for pattern discovery and extrapolation. *30th International Conference on Machine Learning, ICML 2013*, 28(PART 3), 2104–2112, 2013.

[92] J. D. Woloszyn, P. Hesse, K. D. Hungenberg, et K. B. Mcauley. Parameter selection and estimation techniques in a styrene polymerization model. *Macromolecular Reaction Engineering*, 7(7), 293–310, 2013, http://dx.doi.org/10.1002/mren.201200074.

[93] Z. Yang, D. Eddy, S. Krishnamurty, I. Grosse, P. Denno, Y. Lu, et P. Witherell. Investigating Grey-Box Modeling for Predictive Analytics in Smart Manufacturing. (February 2020), 2017, http://dx.doi.org/10.1115/detc2017-67794.

[94] J. Yu, K. Chen, et M. M. Rashid. A Bayesian model averaging based multi-kernel Gaussian process regression framework for nonlinear state estimation and quality prediction of multiphase batch processes with transient dynamics and uncertainty. *Chemical Engineering Science*, 93, 96–109, 2013, http://dx.doi.org/10.1016/j.ces.2013.01.058.

[95] S. Zendehboudi, N. Rezaei, et A. Lohi. Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review. *Applied Energy*, 228(August), 2539–2566, 2018, http://dx.doi.org/10.1016/j.apenergy.2018.06.051.

[96] Y. Zhang et X. Xu. Machine learning glass transition temperature of polymers. *Heliyon*, 6(10), e05055, 2020, http://dx.doi.org/10.1016/j.heliyon.2020.e05055.

[97] Y. Zhang et X. Xu. Machine learning glass transition temperature of polyacrylamides using quantum chemical descriptors. *Polymer Chemistry*, 12(6), 843–851, 2021, http://dx.doi.org/10.1039/d0py01581d.

[98] L. Zhao, Z. Li, Z. Wang, B. Caswell, J. Ouyang, et G. E. Karniadakis. Active- and transfer-learning applied to microscale-macroscale coupling to simulate viscoelastic flows. *Journal of Computational Physics*, 427, 110069, 2021, http://dx.doi.org/10.1016/j.jcp.2020.110069.

[99] P. Zhao, S. Wang, J. Ying, et J. Fu. Non-destructive measurement of cavity pressure during injection molding process based on ultrasonic technology and Gaussian process. *Polymer Testing*, 32(8), 1436–1444, 2013, http://dx.doi.org/10.1016/j.polymertesting.2013.09.006.

[100] M. X. Zhu, Q. C. Yu, H. G. Song, T. X. Chen, et J. M. Chen. Rational Design of High-Energy-Density Polymer Composites by Machine Learning Approach. *ACS Applied Energy Materials*, 4(2), 1449–1458, 2021, http://dx.doi.org/10.1021/acsaem.0c02647.

Résumé étendu en français

Contexte

Les méthodes d'intelligence artificielle (IA) et de machine learning (ML) ont suscité un intérêt croissant au cours des dernières décennies, et le secteur du Génie des Produits (GdP) n'y échappe pas. En effet, ces méthodes basées sur les données, qui se sont beaucoup développées grâce à l'explosion des quantités de données et aux progrès technologiques/mathématiques, sont parvenues à résoudre des problèmes complexes dans d'autres domaines (ex: génération d'images, voitures autonomes, traitement du langage naturel). Par conséquent, elles pourraient s'avérer très intéressantes pour certains défis du GdP, lorsque les approches classiques rencontrent des limites. Par exemple, les méthodes phénoménologiques (i.e., basées sur la description des mécanismes et phénomènes gouvernant un système), lorsqu'elles existent, ne parviennent pas toujours à capturer correctement la relation entre le procédé de fabrication, les structures, les ingrédients et les propriétés d'usage des produits complexes du GdP. En effet, ces derniers sont généralement caractérisés par la multiplicité de leurs structures, propriétés d'utilisation et/ou ingrédients. Un autre exemple concerne la conception et la découverte de nouvelles molécules ou de nouveaux matériaux dotés de propriétés données, qui nécessitent de nouvelles méthodes pour une meilleure exploration du vaste espace chimique. Cependant, le domaine du ML contient une multitude de méthodes et concepts qui sont parfois difficiles ou longs à maîtriser pour les ingénieurs débutants dans ce domaine et désireux d'appliquer ces approches. En outre, le succès des méthodes de ML ne doit pas rendre leur application systématique à n'importe quel problème, car ces méthodes possèdent également leurs propres exigences et limites par rapport aux approches basées sur la connaissance.

Motivations et objectifs

Cette thèse s'intéresse à l'utilisation des approches ML dans le cadre du GdP.

Dans une première partie, le but est de dresser un **état-de-l'art** et de mieux comprendre les caractéristiques, avantages et limites de ces approches et la façon dont elles s'appliquent selon la nature des applications en GdP (ex : quantité de données souvent limitée, représentation complexe des molécules, réactions, spectres ou propriétés sensorielles). Les principaux challenges du ML en GdP y sont également discutés et quelques recommandations sont données pour la sélection d'un modèle ML.

Dans une deuxième partie, l'objectif est de tester différentes approches ML sur deux applications concrètes du GdP où les méthodes classiques s'appliquent difficilement. Celles-ci sont caractérisées par des objectifs et des types de données très différents, impactant donc les approches adoptées.

La **première application** concerne le développement d'un modèle QSPR (relation quantitative structure à propriété) pour prédire deux propriétés thermodynamiques de molécules à partir de leurs caractéristiques structurelles et physico-chimiques. En particulier, les données considérées contiennent deux principaux éléments de complexité. Le premier est la représentation des molécules par un nombre important de descripteurs afin d'augmenter les chances de capturer les caractéristiques pertinentes affectant les propriétés thermodynamiques en question, étant donné l'absence de connaissances suffisantes. Le second est la grande diversité des structures chimiques afin d'élargir l'applicabilité du modèle développé à des fins de découverte

chimique. Surtout, la particularité de cette application réside dans l'approche multi-angle adoptée pour mieux visualiser et comprendre les méthodes possibles à chaque étape (de la collecte des données à la construction du modèle) et leur impact sur le modèle obtenu. En outre, des méthodes permettant de définir le domaine d'applicabilité et de détecter les valeurs aberrantes dans ce problème à haute dimension sont étudiées.

La **deuxième application** se focalise sur la modélisation d'un procédé de polymérisation du styrène en présence de particules de pneus usagés pour prédire le taux de conversion du styrène en fonction des conditions opératoires. Dans ce problème, la quantité de données est beaucoup plus limitée (car les données sont issues d'expériences) mais les mécanismes et la cinétique de certaines parties du système sont bien connus. Ainsi, des méthodes ML et des méthodes hybrides (combinant le ML et les modèles basés sur les connaissances) sont développées et comparées, avec un intérêt particulier pour les Gaussian Processes (GP) pour la partie ML, afin de fournir l'incertitude des prédictions. Le modèle basé sur la connaissance est quant à lui issu d'une précédente thèse et contient la cinétique connue du système.

Ce manuscrit est divisé en 5 chapitres. Le chapitre 1 correspond à l'état de l'art, tandis que les autres chapitres présentent les applications (chapitres 2 et 3 pour la première application et chapitres 4 et 5 pour la deuxième application). Par ailleurs, ce manuscrit est sur articles, ce qui signifie que les chapitres présentés ont été publiés, seront soumis ou feront l'objet d'une publication future.

Démarche et principaux résultats obtenus

Chapitre 1: Etat-de-l'art

Au vu de l'abondance des méthodes de ML et de leurs applications en CPE, ce premier chapitre fournit des éléments d'introduction au ML (ex: caractéristiques et principes de différentes méthodes) tout en donnant une vue d'ensemble sur l'utilisation du ML dans le domaine du GdP, et notamment dans les applications suivantes:

- le design et la découverte de nouvelles molécules et nouveaux matériaux
- la modélisation de procédés
- le support à l'analyse sensorielle
- la prédiction de réactions chimiques

Chacune de ces applications possède des données avec des caractéristiques propres et des problématiques différentes ce qui va influencer le choix des approches ML. En effet, contrairement à d'autres domaines comme la vision par ordinateur ou le traitement du langage naturel qui bénéficient de millions de données sous forme d'images ou textes et qui utilisent donc principalement des méthodes de DL, le domaine du GdP est beaucoup plus limité en termes de données (plusieurs dizaines à milliers dans la plupart des cas) et la représentation des données (ex: molécules, réactions) est un véritable challenge.

L'état-de-l'art met en avant un vrai engouement pour le ML dans le domaine de GdP car il permet de résoudre de nombreuses problématiques telles que la complexité des interactions procédé-propriétés-ingrédients-structures ou la découverte de nouvelles molécules et nouveaux matériaux. En effet, le principal avantage du ML réside dans le fait que ces méthodes peuvent fournir des prédictions très précises et rapides en se basant uniquement sur des données, ce qui est particulièrement intéressant lorsque le comportement d'un système n'est pas connu ou ne peut pas être bien décrit par des équations physico-chimiques.

Néanmoins, les méthodes de ML sont basées sur les données et leurs principales limitations sont donc liées au manque de données (en quantité et en qualité) en GdP, et au manque d'interprétabilité et d'extrapolation de ces modèles. Différentes solutions sont possibles pour y remédier comme l'utilisation de méthodes plus appropriées aux faibles données (ex: transfert learning, active learning) ou la combinaison de ces modèles avec des modèles basés sur la connaissance.

Par ailleurs, il n'existe pas de règles bien définies sur quelle méthode adopter pour chaque problème, cependant il est reconnu que certaines méthodes, du fait de leur construction mathématique notamment, sont plus adaptées à certains types de données/problèmes. Certaines recommandations sont présentées.

Les deux applications étudiées au cours de cette thèse permettent de voir concrètement comment le ML peut s'appliquer à des problèmes aux données et caractéristiques très différentes, où les approches classiques rencontrent des limites.

Chapitres 2 et 3: Première application

Dans cette première application, le but est de développer un modèle ML/QSPR pour prédire deux propriétés thermodynamiques (enthalpie de formation et entropie) de molécules en se basant sur leurs descripteurs, c'est-à-dire leurs caractéristiques structurelles et physico-chimiques. Le travail est divisé en deux parties. La première (Chapitre 2) se concentre sur le développement du modèle QSPR de la collecte des données jusqu'à la construction du modèle (cf. Figure 1). La deuxième (Chapitre 3) s'intéresse à la définition du domaine d'applicabilité du modèle à différents stades du développement du modèle (cf. Figure 2).

Tout d'abord, le recours au ML s'explique ici par le fait que les méthodes classiques pour déterminer ces propriétés (méthode de contributions de groupes, méthodes de chimie quantique) ne peuvent s'appliquer qu'à des molécules simples/petites et requièrent des connaissances expertes pour les appliquer correctement. D'une part, décomposer une molécule en sous-groupes pour lesquels les propriétés thermodynamiques sont connues devient difficile pour des molécules plus complexes/grandes. D'autre part, les méthodes de chimie quantique sont très complexes (car basées sur la résolution de l'équation de Schrödinger) et très lourdes en termes de temps de calcul.

Dans le cadre de ce travail, le développement du modèle QSPR s'effectue sous certains critères/contraintes. En effet, l'objectif final est de pouvoir appliquer le modèle développé à n'importe quelle molécule dans le cadre de la découverte de nouvelles molécules pour diverses applications. Par conséquent, la base de données considérée pour entraîner le modèle est constituée d'une très large variété de structures moléculaires (23 familles). De plus, la déf-

inition du domaine d'applicabilité du modèle est donc cruciale, étant donné qu'un modèle ML n'est pas extrapolable. Une seconde contrainte est le manque de connaissances concernant les caractéristiques structurelles et physico-chimiques pertinentes pour prédire les deux propriétés thermodynamiques. Ainsi, les molécules sont représentées par un large nombre de descripteurs (5666 descripteurs) pour assurer une représentation la plus complète possible. Enfin, le dernier critère se base sur les principes de l'OCDE (Organisation de coopération et de développement économiques) qui ont été établies pour faciliter la considération des modèles QSPR dans la régulation des substances chimiques.

La spécificité de ce travail réside dans l'approche multi-angle adoptée et la définition du domaine d'applicabilité dans le cadre des critères/contraintes énoncées précédemment. Beaucoup de travaux se limitent par exemple à une famille chimique donnée de molécules. D'autres font le choix de descripteurs en se basant sur leur expertise du système et/ou selon une méthode donnée. Aussi, le choix et l'impact des méthodes tout au long du développement des modèles QSPR n'est pas toujours évidente. Enfin, le domaine d'applicabilité est particulièrement difficile à déterminer en haute dimension.

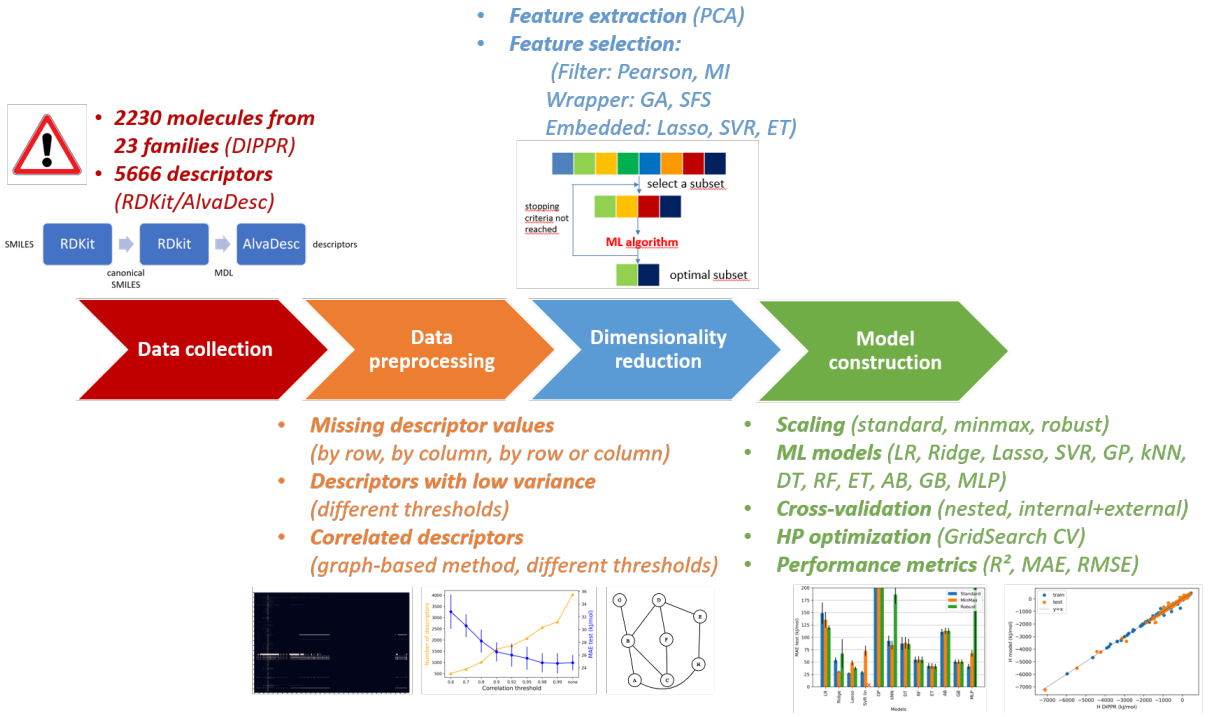


Figure 1: Vue d'ensemble de la méthodologie utilisée dans le Chapitre 2.

L'approche multi-angle adoptée dans le Chapitre 2 a permis de visualiser et comprendre les méthodes possibles à différentes étapes du processus et d'évaluer leur impact sur la performance du modèle. Les méthodes testées sont visibles sur la Figure 1. La configuration finale a été choisie de sorte à ce qu'il y ait un bon compromis entre la précision (faible erreur entre le modèle et les observations), l'interprétabilité (réduction de dimensionalité) et le domaine d'applicabilité (prétraitement n'éliminant pas trop de molécules) du modèle. En particulier, la réduction du nombre de descripteurs par un algorithme génétique et l'entraînement d'un modèle *Lasso* sur les données obtenues ont permis d'obtenir un bon compromis, avec un MAE de test de 24.2 kJ/mol pour l'enthalpie de formation et 17.4 J/mol/K pour l'entropie. Par ailleurs, les descripteurs

identifiés se sont avérés physiquement cohérents. Enfin, un benchmark a été effectué et a montré des performances comparables entre le modèle développé et ceux de la littérature.

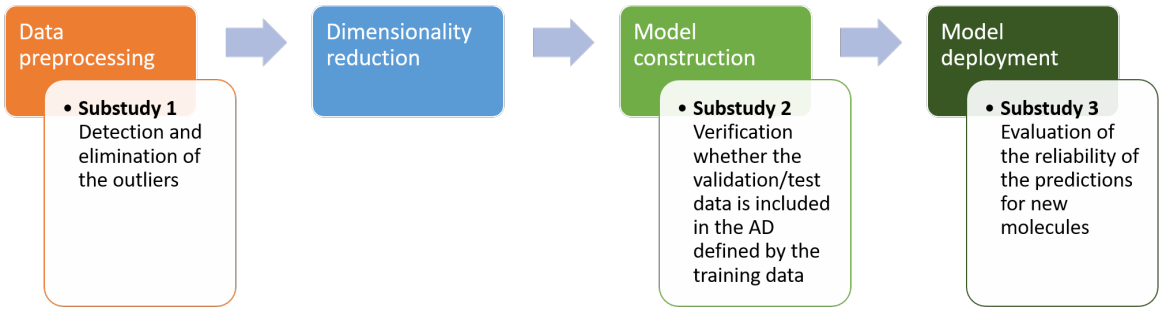


Figure 2: Vue d'ensemble de la méthodologie utilisée dans le Chapitre 3.

Pour savoir si le modèle peut être appliqué à une nouvelle molécule donnée, le domaine d'applicabilité du modèle a ensuite été étudié dans le Chapitre 3. Le domaine d'applicabilité contient l'ensemble des structures chimiques et propriétés pour lesquelles un modèle est capable d'effectuer des prédictions avec une certaine fiabilité. En général, la plupart des méthodes existantes pour le définir s'appliquent mieux en faible dimension (i.e., inférieur à 10 alors que l'algorithme génétique a sélectionné 100 descripteurs au chapitre précédent) et sont utilisées au moment de la construction du modèle pour vérifier que les données de validation ou test sont bien situées dans le domaine défini par les données d'entraînement. Ici, différentes méthodes, plus adaptées en haute dimension, ont été étudiées à différentes étapes du développement du modèle QSPR (cf. Figure 2).

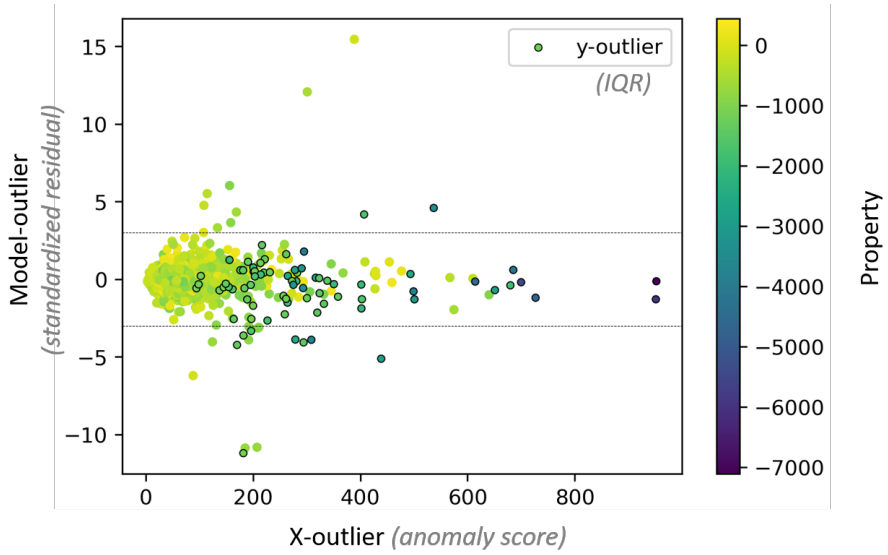


Figure 3: Types d'outliers.

Pendant la phase de prétraitement des données, définir le domaine d'applicabilité revient à éliminer les données aberrantes (ou outliers). Un outlier est un point qui présente un comportement très différent par rapport aux autres points d'un même ensemble. De ce fait, il existe différents types d'outliers (cf. Figure 3): les outliers dans l'espace des descripteurs (X-

outliers), ceux dans l'espace des réponses (y-outliers) et ceux dans l'espace des prédictions du modèle (Model-outliers, les points qui présentent de fortes erreurs de prédiction). Les deux dernières catégories sont unidimensionnelles et des méthodes univariées ont donc été utilisées pour les détecter (ex: boxplots pour les y-outliers). En revanche, l'espace des descripteurs étant de haute dimension, trois méthodes de détection des X-outliers ont été comparées: *Isolation Forest*, l'écart-type des prédictions dans *Random Forest* et *k-nearest neighbors* dans l'espace des descripteurs projeté en 2D via *t-SNE* (t-distributed stochastic neighbor embedding). Après avoir testé différentes méthodes d'élimination d'outliers, avec différents seuils et ordres, la configuration retenue a permis de diviser le MAE par 2 et le RMSE par 3 tout en réduisant le phénomène d'overfitting et limitant la perte de données (cf. Tableau 1).

Table 1: Impact de l'élimination des outliers sur la performances des modèles pour la prédiction de l'enthalpie (kJ/mol) et de l'entropie (J/mol/K).

Propriété	**Elimination des outliers**	**Mol.**	**Desc.**	R^2 **train**	R^2 **test**	MAE **train**	MAE **test**	$RMSE$ **train**	$RMSE$ **test**
Enthalpie	Non	1785	100	0.995	0.978	15.45	24.16	36.90	70.77
	Oui	1531	100	0.998	0.994	9.27	11.89	14.09	21.50
Entropie	Non	1747	100	0.980	0.969	14.09	17.37	29.23	35.49
	Oui	1514	100	0.996	0.995	7.09	8.01	9.78	11.49

Durant la construction du modèle, la définition du domaine d'applicabilité a pour but de vérifier que les données de validation et de test sont bien incluses dans le domaine d'applicabilité défini par les données d'entraînement. Bien que pour le modèle développé, la majorité des données de test étaient bien situées proches des données d'entraînement, il reste difficile de définir clairement les limites du domaine d'applicabilité du modèle (i.e., valeurs seuils selon chaque axe sur la Figure 3). En effet, les propriétés de certaines molécules éloignées des données d'entraînement sont parfois mieux prédites que pour certaines molécules plus proches.

Enfin, lors de l'utilisation du modèle sur des nouvelles molécules, le domaine d'applicabilité permet d'évaluer la fiabilité des prédictions. Du fait du problème soulevé précédemment, il est pour le moment difficile de conclure sur la fiabilité des prédictions pour les nouvelles molécules étudiées dans cette dernière partie.

Chapitres 3 et 4: Deuxième application

Cette deuxième application a pour objectif de modéliser le système de polymérisation radicalaire du styrène en présence de particules de pneus usagés (aussi appelées GTR, ground tire rubber). L'intérêt de cette réaction est de permettre le recyclage des pneus usagés tout en créant, par un mécanisme de greffage, des matériaux composites polystyrène/GTR avec des propriétés mécaniques améliorées par rapport à du polystyrène pur. Pour que cette réaction soit optimale, il faut à la fois maximiser le taux de conversion du styrène et l'efficacité de greffage du polystyrène sur le GTR. Ce travail se focalise sur le développement de modèles ML pour prédire le taux de conversion en fonction des conditions opératoires. Cette deuxième application est très différente de la première au niveau de données et donc des approches utilisées: il y a beaucoup moins de données (car générées expérimentalement) mais elles sont de faible dimension (conditions opératoires). De plus, le modèle cinétique d'une partie du système est disponible.

Dans ce problème, l'utilisation du ML se justifie du fait que la modélisation de la cinétique de ce système par des équations physico-chimiques est particulièrement complexe à cause de la présence du GTR. En effet, ce dernier a une structure 3D et une composition complexes. Etant issu de différents types de pneus, la composition exacte du GTR (types/proportions d'élastomères et d'additifs) n'est pas connue ce qui ne permet pas une description précise des phénomènes et réactions qui entrent en jeu entre les différentes espèces présentes. Le GTR et notamment l'un de ces principaux additifs, le noir de carbone, peuvent justement impacter le cours de la réaction de façon significative.

La première partie de ce travail (Chapitre 4) se concentre le développement de modèles ML "purs", c'est-à-dire non combinés avec des modèles cinétiques. Dans un premier temps, une procédure expérimentale a été établie afin de produire les données nécessaires au modèle de ML. Une attention particulière a été portée sur la quantité et la qualité des données. En effet, le protocole défini permet de réaliser plusieurs réactions en parallèle, pour produire davantage de données. De plus, pour assurer la qualité des données, des faibles volumes ont été utilisés afin d'analyser la totalité du contenu des réacteurs et donc éviter les incertitudes liées à l'échantillonnage dans ce système hétérogène, et les étapes ont été bien contrôlées (répertoriage des pertes, incertitudes de mesures, répétition de certains points, contrôle de la température). Deux échelles expérimentales ont été considérées (10 mL et 100 mL) et la première a été retenue, car le répertoriage des pertes était plus fiable et la température plus homogène au sein du système. Enfin, les données expérimentales obtenues ont été utilisées pour entraîner différents modèles de ML. Les meilleures performances ont été obtenues pour les modèles *Gradient Boosting*, *Multilayer perceptron* et *Random Forest*, avec des R^2 test de 0.91, 0.90 et 0.89 respectivement.

Table 2: Comparaison des modèles white-box (WB), black-box (BB) et hybride.

Modèles	**Interprétabilité physique**	**Capacité d'extrapolation**	**Temps de calcul**	**Complexité du modèle**	**Données requises**
WB	Bon	Bon	Elevé	Elevé	Faible
BB	Mauvais	Mauvais	Faible	Faible	Elevé
Hybride	Intermédiaire	Intermédiaire	Intermédiaire	Intermédiaire	Intermédiaire

Dans une seconde partie (Chapitre 5), des approches hybrides combinant des modèles ML (ou black-box, BB) et des modèles basés sur les connaissances (ou white-box, WB) sont étudiées. L'avantage des approches hybrides est d'exploiter toute l'information disponible sur le système, c'est-à-dire les données et les connaissances, afin d'avoir un modèle plus performant et fiable. Plus précisément, le modèle hybride permet de contrebalancer les avantages et inconvénients des modèles WB et BB, comme le montre le Tableau 2. D'un point de vue pratique, la méthode *Gaussian Processes* est utilisée pour la partie ML pour sa capacité à fournir les incertitudes de prédiction, ce qui plus réaliste pour le système étudié. Par ailleurs, un modèle cinétique établi lors d'une thèse précédente a été considérée pour la partie WB. Ce modèle a été construit de sorte à décrire de la façon la plus détaillée possible tous les phénomènes et réactions connus pour ce système. Les modèles WB et BB sont combinés selon les schémas présentés sur les Figures 4 et 5. Pendant la phase de construction du modèle (Figure 4), le modèle ML est entraîné pour prédire l'écart entre les prédictions du modèle WB et les données expérimentales. Pendant la phase d'application du modèle (Figure 5), les prédictions des deux modèles sont additionnées afin d'obtenir la prédiction finale. L'approche hybride a été testée sur différentes données et a donné des modèles plus précis, physiquement réalistes, interprétables et extrapolables, comme dans le cas présenté sur la Figure 6. Plusieurs pistes ont également été proposées afin d'améliorer

ce modèle préliminaire.

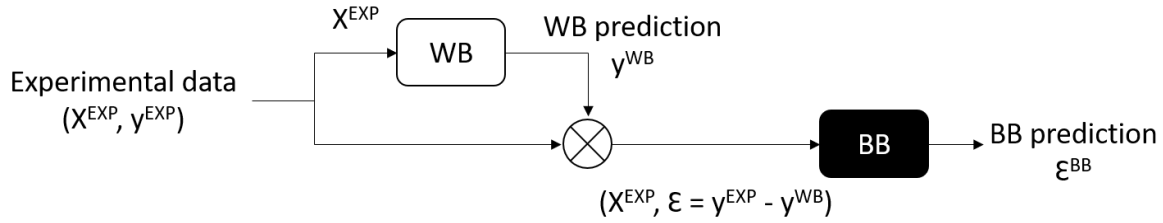


Figure 4: Construction du modèle hybride

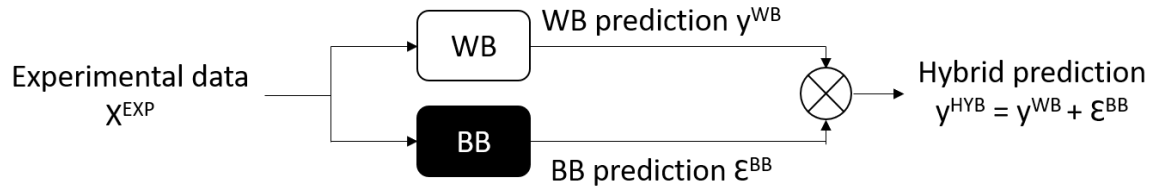


Figure 5: Application du modèle hybride

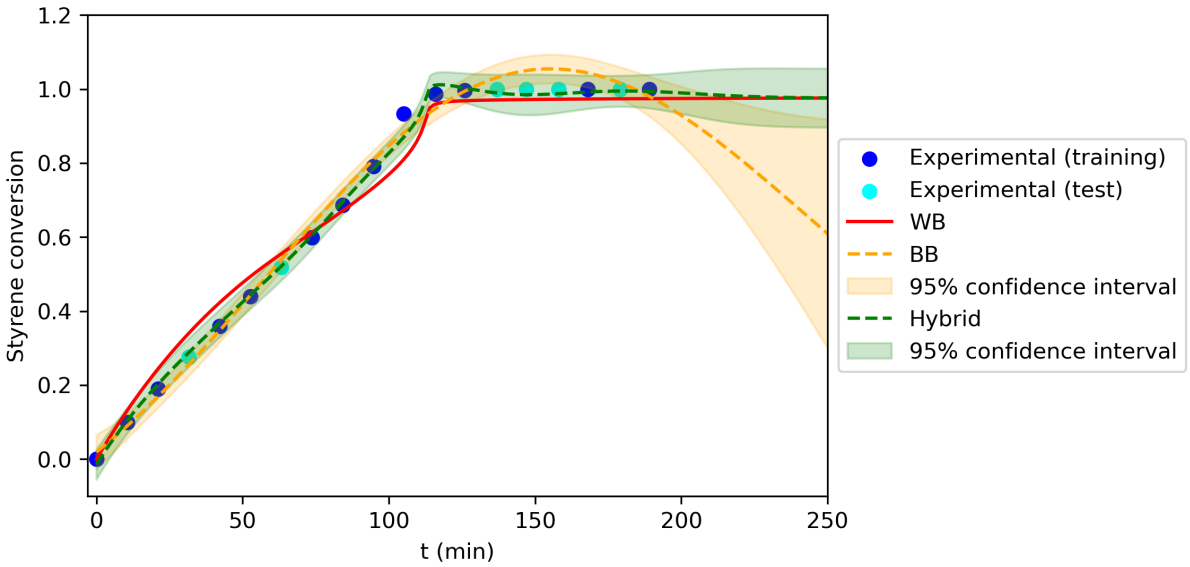


Figure 6: Comparaison des prédictions des modèles WB, BB et hybride pour un cas d'étude.

Conclusions générales et perspectives

L'apport positif des méthodes de ML pour la résolution de nombreux problèmes en GdP est évident. En effet, beaucoup de travaux dans la littérature et ceux effectués dans le cadre de cette thèse ont donné des résultats prometteurs sur des problèmes où les approches classiques ne sont pas applicables. Cependant, de nombreux challenges ont aussi pu être identifiés, à travers l'état-

de-l'art et les deux applications étudiées. Ces challenges sont principalement caractéristiques du domaine du GdP.

Tout d'abord, la **représentation des données chimiques** (ex: molécules, réactions, spectres, propriétés sensorielles) est un domaine de recherche actif. Dans le premier cas d'étude de cette thèse, les molécules ont été représentées par des descripteurs et les modèles construits ont donné de bonnes performances par rapport aux autres représentations et modèles développés dans la littérature. Néanmoins, chaque type de représentation possède ses avantages et inconvénients et leur choix dépendra donc de chaque problème. Une piste d'amélioration pour le premier cas d'étude se situe donc au niveau de la représentation moléculaire (amélioration de la représentation par des descripteurs ou test d'autres types de représentations comme les graph neural networks).

Un autre axe de recherche important concerne le **développement des modèles hybrides** qui combinent les approches ML avec les approches basées sur les connaissances existantes. Ce type de modèle est intéressant pour le domaine de GdP, où une connaissance partielle des systèmes et une faible quantité de données sont souvent disponibles, car elle permet de pallier aux principaux problèmes du ML (i.e., manque d'interprétabilité, mauvaise extrapolabilité, nécessité d'une grande quantité de données) tout en évitant le développement complexe de modèles basés sur la connaissance. La façon d'intégrer les connaissances reste cependant un point à améliorer, comme dans la deuxième application de cette thèse où une seule configuration hybride a pu été testée. Une approche hybride pourrait également être intéressant pour améliorer l'interprétabilité et l'extrapolabilité des modèles QSPR développés dans la première application. Plus généralement, la tendance est à la recherche de méthodes pour améliorer l'**interprétabilité** des modèles ML, comme le montrent plusieurs études récentes [23, 25, 59, 61].

La **disponibilité des données** est un autre point critique dans certaines applications du GdP, par exemple lorsqu'elles sont générées expérimentalement, lorsqu'un système a été peu étudié ou à cause de problèmes de confidentialité/compétitivité. Pour cette raison, de nombreuses approches ML, plus adaptées à un faible nombre de données, font l'objet de recherches tels que l'apprentissage par transfert (transfer learning), l'apprentissage actif (active learning), l'apprentissage semi-supervisé ou les modèles hybrides. De plus, le partage des données scientifiques est de plus en plus encouragé au sein de la communauté scientifique. Ces données incluent les résultats positifs (et négatifs) ainsi que toutes les informations sur la méthode utilisée pour générer les données et leurs incertitudes. Une meilleure disponibilité des données permettrait également de réaliser des benchmarks plus facilement.

Par ailleurs, les éléments suivants méritent aussi d'être étudiés davantage. Comme dans les deux applications de cette thèse, les données en GdP sont souvent accompagnées d'**incertitudes** qu'il serait intéressant de pouvoir intégrer correctement aux modèles ML pour obtenir des modèles plus généralisables. De la même façon, accompagner les prédictions des modèles ML de leurs incertitudes permettrait d'avoir des prédictions plus représentatives de la réalité physique. En ce sens, les Gaussian Processes étudiés dans la deuxième application représentent une option possible.

Un aspect souvent peu étudié mais important concerne le **domaine d'applicabilité** des modèles, comme le démontre le premier cas d'étude. Dans le cadre des modèles QSPR/QSAR, cet aspect gagne de l'importance afin d'améliorer leur utilisation dans la réglementation des produits chimiques entre autres. Cependant, des approches plus adaptées sont nécessaires, en

particulier en haute dimension.

D'une manière plus générale, le ML possède ses propres limites et critères, dont il faut être conscient avant de commencer un projet ML. De plus, le développement d'un modèle ML ne consiste pas à seulement entraîner un modèle sur des données et avoir une bonne performance. Au contraire, un projet ML est composé de plusieurs étapes, toutes très importantes. La collection et la préparation des données en quantité et qualité est un premier défi. Puis, beaucoup de méthodes possibles existent à chaque étape du développement du modèle selon les caractéristiques des données et chaque choix effectué impacte le modèle final. Il n'existe cependant pas de guide précis donnant la démarche à adopter pour chaque problème mais les cas d'études proposés dans cette thèse montrent des méthodes possibles pour différents problèmes.

Enfin, il est certain que les recherches futures permettront au fur et à mesure de mieux appréhender l'utilisation des approches ML, et d'exploiter tout le potentiel de ces méthodes pour trouver des solutions adaptés aux défis complexes du GdP.

Résumé & Abstract

Résumé

Les méthodes d'intelligence artificielle (IA) et de machine learning (ML) ont suscité un intérêt croissant au cours des dernières décennies, et le secteur du Génie des Produits (GdP) n'y échappe pas. En effet, ces méthodes basées sur les données, qui se sont beaucoup développées grâce à l'explosion des quantités de données et aux nombreux progrès techniques, sont parvenues à résoudre des problèmes complexes dans d'autres domaines. Par conséquent, elles pourraient s'avérer très intéressantes pour certains défis du GdP lorsque les approches classiques rencontrent des limites. Par exemple, les méthodes phénoménologiques (i.e., basées sur la description des mécanismes et phénomènes gouvernant un système), lorsqu'elles existent, ne parviennent pas toujours à capturer correctement le lien entre le procédé de fabrication, les structures et les propriétés d'usage des produits complexes du GdP.

Cette thèse s'intéresse à l'utilisation des approches ML dans le cadre du GdP. Dans une première partie, le but est de dresser un état-de-l'art et de mieux comprendre les caractéristiques, avantages et limites de ces approches et la façon dont elles s'appliquent selon la nature des applications en GdP (ex : quantité de données limitée, représentation moléculaire complexe). Les principaux challenges du ML en GdP y sont également discutés. Dans une deuxième partie, l'objectif est de tester différentes approches ML sur deux applications concrètes du GdP. Celles-ci sont caractérisées par des objectifs et des types de données très différents, impactant donc les approches adoptées. La première application concerne le développement d'un modèle QSPR (relation quantitative structure à propriété) pour prédire deux propriétés thermodynamiques de molécules à partir de leurs caractéristiques structurelles et physico-chimiques. La particularité de cette application est l'approche multi-angle adoptée pour mieux visualiser et comprendre les méthodes possibles à chaque étape et leur impact sur le modèle obtenu. La deuxième application se focalise sur la modélisation d'un procédé de polymérisation du styrène en présence et absence de particules de pneus usagés pour prédire le taux de conversion du styrène en fonction des conditions opératoires. Des méthodes ML et des méthodes hybrides (combinant ML et modèle de connaissance) y sont développés, avec un intérêt particulier pour les Gaussian processes pour la partie ML afin de fournir l'incertitude des prédictions.

Mots-clés : Intelligence artificielle/Machine learning, Génie des produits, QSPR/QSAR, Propriétés thermodynamiques, Polymérisation, Modélisation mathématique

Abstract

Artificial intelligence (AI) and machine learning (ML) methods have gained growing interest in recent decades, and the Chemical Product Engineering (CPE) sector is not exempt. Indeed, these data-driven methods, which have developed greatly thanks to the explosion of data availability and to various technical advances, have succeeded in solving complex problems in other fields. As a result, they could be very interesting for certain CPE challenges, when classical approaches reach their limits. For example, phenomenological methods (i.e., based on the description of the mechanisms and phenomena that drive a process), when they exist, do not always manage to correctly capture the link between the manufacturing process, structures and usage properties of complex CPE products. At the same time, the ML field contains a plethora of methods, sometimes difficult to master for newcomers, and these methods have certain limits.

This thesis focuses on the use of ML approaches in the context of CPE. In the first part, the aim is to provide an overview and a better understanding of the characteristics, advantages and drawbacks of these approaches and how they apply depending on the nature of CPE applications (e.g. limited amount of data, complex molecular representation). The main challenges of ML in CPE are also discussed. In the second part, the goal is to test different ML approaches on two concrete CPE applications. These are characterized by very different objectives and data characteristics, thus impacting the adopted approaches. The first application concerns the development of a QSPR (quantitative structure-property relationship) model to predict two thermodynamic properties of molecules from their structural and physico-chemical characteristics. The special feature of this application lies in the multi-angle approach adopted to better visualize and understand the possible methods at each stage and their impact on the model obtained. The second application deals with the modeling of a styrene polymerization process in presence and absence of used tire particles, in order to predict the styrene conversion rate as a function of operating conditions. Both ML methods and hybrid ML/knowledge-based methods are developed, with a particular interest in Gaussian processes for the ML part, to provide uncertainty in predictions.

Keywords : Artificial intelligence/Machine learning, Chemical product engineering, QSPR/QSAR, Thermodynamic properties, Polymerization, Mathematical modeling