



# Hybrid modeling of hetero-agglomeration processes: a framework for model selection and arrangement

Frank Rhein<sup>1</sup> · Leonard Hibbe<sup>1</sup> · Hermann Nirschl<sup>1</sup>

Received: 14 September 2022 / Accepted: 14 March 2023  
© The Author(s) 2023

## Abstract

Modeling of hetero-agglomeration processes is invaluable for a variety of applications in particle technology. Traditionally, population balance equations (PBE) are employed; however, calculation of kinetic rates is challenging due to heterogeneous surface properties and insufficient material data. This study investigates how the integration of machine learning (ML) techniques—resulting in so-called hybrid models (HM)—can help to integrate experimental data and close this gap. A variety of ML algorithms can either be used to estimate kinetic rates for the PBE (serial HM) or to correct the PBE's output (parallel HM). As the optimal choice of the HM architecture is highly problem-dependent, we propose a general and objective framework for model selection and arrangement. A repeated nested cross-validation with integrated hyper-parameter optimization ensures a fair and meaningful comparison between different HMs. This framework was subsequently applied to experimental data of magnetic seeded filtration, where prediction errors of the pure PBE were reduced by applying the hybrid modeling approach. The framework helped to identify that for the given data set, serial outperforms parallel arrangement and that more advanced ML algorithms provide better interpolation ability. Additionally, it enables to draw inferences to general properties of the underlying PBE model and a statistical investigation of hyper-parameter optimization that paves the way for further improvements.

**Keywords** Hetero-agglomeration · Population balance equations · Machine learning · Hybrid modeling · Hyper-parameter optimization · Repeated nested cross-validation

## 1 Introduction

Agglomeration refers to the assembly of particles into larger clusters (the agglomerates). It is a unit operation of mechanical process engineering and thus relevant for almost all processes in particle technology. Homo-agglomeration of a single particle system is well understood and mainly used to either formulate products from powders (tablets) or to improve separation properties in suspensions (flocculation). Hetero-agglomeration refers to agglomeration processes

between different materials and results in multi-substance compounds. These hetero-agglomerates often combine desired properties of the individual materials and therefore exhibit a wide range of applications: In cathode materials for lithium-ion batteries, carbon black is agglomerated with the active material to enhance conductivity. The hetero-agglomeration process further defines the micro-structure of these cathode materials and significantly impacts the final product properties [44]. Throughout this work, magnetic seeded filtration (MSF) [62–64] is used as application example, where the separation behavior of suspensions is drastically changed by hetero-agglomeration: Non-magnetic target particles are selectively agglomerated with magnetic seed particles and the formed hetero-agglomerates are subsequently magnetically separated due to their newly gained magnetic properties.

On an experimental level, a “simple” measurement of the agglomerate size is often not sufficient, since many product properties only arise from the micro-structure of the hetero-agglomerates. Meanwhile, only limited analytical

✉ Frank Rhein  
frank.rhein@kit.edu

Leonard Hibbe  
leonard.hibbe@gmx.net

Hermann Nirschl  
hermann.nirschl@kit.edu

<sup>1</sup> Institute of Mechanical Process Engineering and Mechanics, Karlsruhe Institute of Technology (KIT), Strasse am Forum 8, 76131 Karlsruhe, Germany

capabilities exist to determine sought-after properties, such as the material-specific agglomerate composition, most of which are laborious and offline. This motivates the search for models that provide insight into the micro-processes and kinetics. Usually, population balance equations (PBE) with different numerical solution techniques are employed; however, calculations are challenging, because all interactions between all components must be taken into account and formed hetero-agglomerates have locally heterogeneous surface properties. Generally, the accurate calculation of agglomeration rates or so-called kernels pose the limiting factor. Mechanistic equations exist, which, however, are subject to specific assumptions. Furthermore, the quality and quantity of underlying material and process data is unsatisfactory: literature values of HAMAKER constants are off by orders of magnitude due to surface roughness, and experimentally determined zeta potentials are by definition only a makeshift for the sought surface potentials. This multitude of uncertainties makes a predictive and reliable calculation of hetero-agglomeration processes highly challenging [55].

Machine learning (ML) methods are a promising pathway for introducing experimental data into the calculations. They have gained significant traction over the last decade and are becoming a standard modeling technique in chemical engineering [86]. ML methods generally map any given input space on a desired output space and are referred to as black-box models (BBM) throughout this work due to being empirical in nature. This already poses the major limitation for a direct application in hetero-agglomeration processes: As the name suggests, they do not give any insights into the physical micro-processes and generally possess poor extrapolation qualities. A promising solution are the so-called hybrid models (HM) that employ a mechanistic or white-box model (WBM), as e.g., the PBE mentioned above and utilize BBMs to perform certain sub-tasks [87]. HMs can be arranged in various ways: BBM and WBM may be operated in parallel, where the BBM corrects the output of the WBM, or in serial, where typically the BBM calculates and passes necessary kinetic parameters to the WBM [87]. Furthermore, there is a variety of ML algorithms that can be used to realize the BBM, resulting in a wide range of resulting HM structures.

Early examples for the application of HM in chemical engineering date back to the 1990 s [56, 78], while current applications are summarized in von Stosch et al. [87], Sansana et al. [65], Sharma and Liu [72], Mowbray et al. [49], McBride et al. [45]; Zendejboudi et al. [93], and Thon et al. [79] and include general process modeling [2, 35, 58], process control [22, 66, 91], process optimization [5, 54, 94], the development of soft sensors [24, 31, 90], and scale-up [46, 47, 73]. HMs have also been used in combination with PBE to model agglomeration processes: Georgieva et al. [21] employed a series connection of ANN and PBE

to model batch crystallization. Similarly, Hornik et al. [27] modeled batch crystallization of cobalt oxalate with a parallel arrangement. Recently, Dosta and Chan [15] combined micro-scale DEM simulations for retrieving mechanical properties with a hybrid ANN/PBE model, ultimately realizing a multi-scale technique. Nazemzadeh et al. [50] and Nielsen et al. [51] applied an HM for modeling a flocculation process and the development of a soft sensor.

The generalization and extrapolation capabilities of a HM are directly linked to its structure. This encompasses both the used ML algorithm as well as the arrangement of BBM and WBM in serial or parallel [87]. Several studies are concerned with a comparison between the general model structure, without varying the ML algorithm: They compare serial to parallel arrangement [6, 13, 40], multiple more elaborate arrangements [7, 43], or different HMs to both the pure WBM and BBM [83]. Other studies focus on the comparison of different ML algorithms in either a pure BBM [60], a parallel [41], or serial [34, 48, 89] hybrid arrangement. There are only very few studies that focus on a combined investigation of ML algorithm and HM structure. Chen and Ierapetritou [11] investigate the pure WBM, pure BBM, and HMs in serial and parallel while varying the ML algorithm between an artificial neural network (ANN) and a support vector machine (SVM). As the optimal combination is generally problem-dependent, there is no best or one size fits all HM model [87], which at first glance makes comparing different HM structures redundant. However, this emphasizes the necessity for a general framework that enables a problem-specific comparison, optimization, and selection.

The aforementioned points give rise to three distinct goals that we address in this study:

1. Present a general approach for hybrid modeling of hetero-agglomeration processes with PBE.
2. Present a general framework that allows for a fair and problem-specific comparison of various HMs that includes both the HM structure (serial or parallel) and the used ML algorithm.
3. Apply this framework to magnetic seeded filtration and discuss how it can be generalized to enhance hybrid model performance in a variety of chemical engineering problems.

## 2 Models

### 2.1 White-box model: discrete 2-D population balance

Population balance equations (PBE) are the standard technique for calculating the time-dependent number concentrations  $n(x, y, t)$  of agglomerates with properties  $x$  and  $y$

(2-D case). As this work is focused on MSF as application example,  $x$  vividly describes the partial volume of the non-magnetic component (NM) and  $y$  the partial volume of the magnetic component (M), so that every conceivable hetero-agglomerate build from these materials can be fully characterized with a pair of values  $(x, y)$ . It should be emphasized that the presented framework can readily be generalized to higher dimensional problems and that the 2-D case is merely a working example. By defining  $x$  and  $y$  as volumes, the additivity is directly given by mass conservation and the general form of the continuous PBE is given by Ramkrishna [57]

$$\frac{\partial n(x, y, t)}{\partial t} = \underbrace{\frac{1}{2} \int_0^x \int_0^y k(x-x', y-y', x', y') n(x-x', y-y', t) n(x', y', t) dx' dy'}_{(A)} - \underbrace{\int_0^\infty \int_0^\infty k(x, y, x', y') n(x, y, t) n(x', y', t) dx' dy'}_{(B)} \quad (1)$$

Note that Eq. 1 neglects both breakage and convective transport effects. Term (A) describes the formation rate of  $(x, y)$  from agglomeration processes of smaller classes. Since the internal coordinates (volumes) are additive,  $(x, y)$  results from the agglomeration of any class  $(x', y')$  (with  $x' < x$  and  $y' < y$ ) and its associated partner  $(x - x', y - y')$ . The kinetic rates or agglomeration kernels  $k$  are elaborated in Eqs. 4 and 5. Term (B) has a negative sign and describes the loss of  $(x, y)$  due to further agglomeration. Since  $(x, y)$  is in principle capable of agglomerating with all possible classes  $(x', y')$ , integration is performed over the entire range of values.

Equation 1 is an integro-differential equation and can only be solved analytically for simplified cases, which is why a numerical consideration of the problem is necessary. One way of doing so, is to transfer the problem from the continuous space  $(x, y)$  to the discrete space  $(i, j)$ , which transforms Eq. 1 into a set of ODEs. This work employs the so-called geometric discretization scheme, where the agglomerate volume along a coordinate increases by a constant factor  $s$  and which is schematically shown in Fig. 1 for  $s = 2$ . Starting from primary particles  $(i, j) = (1, 0)$  for NM and  $(i, j) = (0, 1)$  for M, all conceivable agglomerate combinations up to a critical agglomerate size defined by the lattice parameter  $N_s$  are defined according to

$$V(i, j) = V_{NM}(i, j) + V_M(i, j) \quad (2)$$

$$V_{NM}(i, j) = s^{i-1} V(1, 0) \quad | \quad V_M(i, j) = s^{j-1} V(0, 1). \quad (3)$$

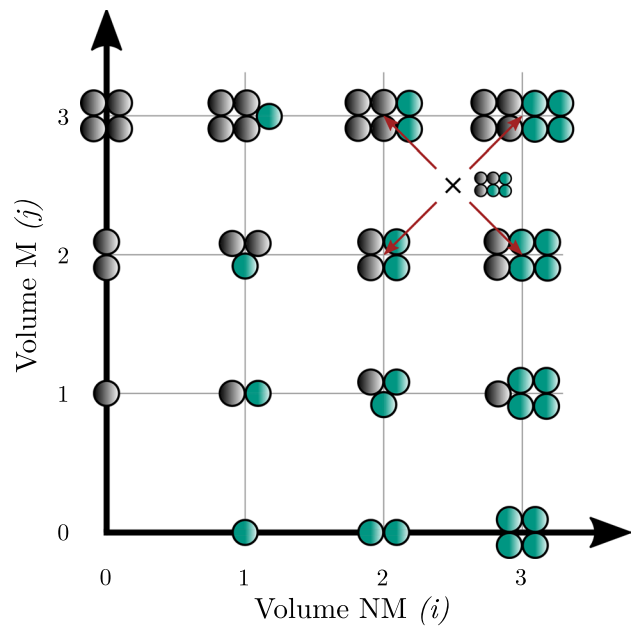


Fig. 1 Discretization with geometric grid.  $N_s = 3$  and  $s = 2$ . The agglomerate of  $(1, 2)$  and  $(2, 1)$  has to be distributed between neighboring grid points

This type of discretization makes calculations manageable for poly-disperse particle systems which would require orders of magnitude more grid points in equidistant grids. However, since the spatial discretization is based on a factor  $s$ , not all formed agglomerates fall on exactly one grid point. As a consequence, agglomerate birth [Term (A) in Eq. 1] cannot always be accounted for directly, but must first be distributed in some way among the surrounding grid points. Figure 1 illustrates this effect for the agglomerate of  $(1, 2)$  with  $(2, 1)$ . Kumar et al. [38] developed the cell average technique, which performs said distribution while ensuring mass conservation in the system. Due to their lengthy and engineer-detering form, the modified birth term and resulting ODEs are given in Appendix 1.

The agglomeration kernels  $k$  are required between each of the investigated agglomerate classes. They are defined as the product of collision frequency  $\beta$  and collision efficiency  $\alpha$  ( $k = \beta\alpha$ ) and are calculated depending on the applied model assumptions which are reviewed, e.g., by Jeldres et al. [32], Taboada-Serrano et al. [76] or Elimelech [16]. In the orthokinetic (flow-controlled) case, the collision frequency is calculated according to Eq. 4 and is proportional to the agglomerate volume (radius  $r^3$ ) and mean shear rate  $\bar{G}$  [12, 16]

$$\beta_{12} = \frac{4}{3} (r_1 + r_2)^3 \bar{G}. \quad (4)$$

Defining the probability of agglomeration upon collision, the collision efficiency generally requires an analysis of particle

trajectories in the orthokinetic case. However, Selomulya et al. [68] presented an empirical factor that corrects the perikinetic  $\alpha^*$ -values obtained by the modified Fuchs approach [20, 26]. This integrates size effects into the PBE while drastically reducing computational effort. The definition and incorporation of this factor is shown in Eq. 12. The perikinetic collision efficiency is calculated according to

$$\alpha_{12}^* = \left[ 2 \int_0^\infty \frac{6u^2 + 13u + 2}{6u^2 + 4u} \exp\left(\frac{E_{el,12} + E_{vdW,12}}{k_B T}\right) \frac{du}{(u + 2)^2} \right]^{-1}, \tag{5}$$

with  $u$  being the dimensionless form of surface-to-surface distance  $h$  according to

$$u(h) = \frac{2h}{r_1 + r_2}. \tag{6}$$

The collision efficiency is governed by the interplay between attractive and repulsive particle–particle interactions. This work is only concerned with electrostatic and van der Waals (DLVO) interactions, while other applications might require the integration of, e.g., hydrophobic interaction energies. The electrostatic interaction energy  $E_{el,12}$  is defined as [23]

$$E_{el,12}(h) = \frac{128\pi r_1 r_2 N_A k_B T}{(r_1 + r_2) \kappa^2} \gamma_1 \gamma_2 \exp(-\kappa h) \tag{7}$$

$$\kappa^{-1} = \sqrt{\frac{\epsilon k_B T}{2e^2 N_A}} \quad \left| \quad \gamma_i = \tanh\left(\frac{e\zeta_i}{4k_B T}\right) \right. \tag{8}$$

with reciprocal DEBYE length  $\kappa$ , permittivity  $\epsilon$ , electron charge  $e$ , and ionic strength  $I$ . The van der Waals interaction energy  $E_{vdW,12}$  is given by [25]

$$E_{vdW,12}(h) = -\frac{A_{H,12}}{6} \left[ \frac{2r_1 r_1}{h^2 + 2r_1 h + 2r_2 h} + \frac{2r_1 r_2}{h^2 + 2r_1 h + 2r_2 h + 4r_1 r_2} + \ln\left(\frac{h^2 + 2r_1 h + 2r_2 h}{h^2 + 2r_1 h + 2r_2 h + 4r_1 r_2}\right) \right]. \tag{9}$$

Calculating the interaction energies in Eqs. 7 and 9 requires knowledge of material-specific surface properties in form of the zeta potential  $\zeta$  and the HAMAKER constant  $A_H$ . This poses a problem in the case of hetero-agglomerates: How is the integral property of a heterogeneous surface defined? A previous publication solved this issue by introducing the so-called collision case model [61] that is briefly elaborated in the following. Figure 2 shows the collision of two hetero-agglomerates  $(i, j)$  and  $(l, m)$ . Depending on the surface property with which each agglomerate collides, four

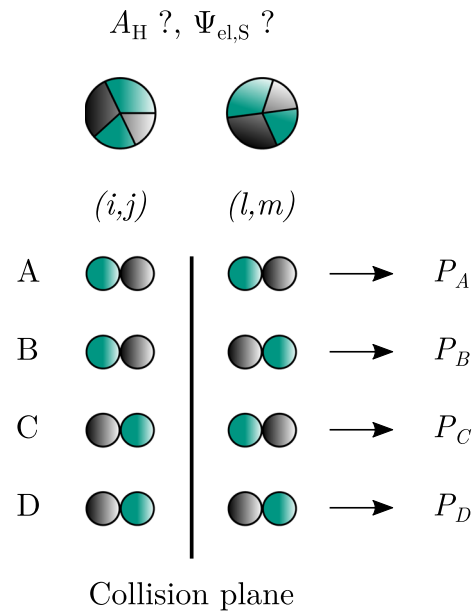


Fig. 2 Definition of possible collision cases A–D between two hetero-agglomerates  $(i, j)$  and  $(l, m)$

collision cases (A) to D) can be differentiated. Vividly, in case A),  $(i, j)$  collides with its magnetic and  $(l, m)$  collides with its non-magnetic side. Since the material properties in the collision plane are defined, a collision efficiency  $\alpha_{A-D}^*$  can be calculated according to Eq. 5 for each case. Subsequently, the probabilities of each case are required. They are defined as the product of the individual probabilities of the agglomerates colliding with the respective material. The latter are estimated according to Eq. 10 under the assumption that the partial volumes are perfectly distributed and that no preferred orientation is present

$$P_{ij,M} = 1 - P_{ij,NM} = X_M(i, j) = \begin{cases} s^{j-1} V(0, 1) & j > 0 \\ 0 & j = 0. \end{cases} \tag{10}$$

It is apparent that the required probabilities are merely a function of the agglomerate volume composition, which is pre-defined by the implemented discretization scheme. Combining all cases, the collision efficiency of the overall collision is calculated according to

$$\alpha^*(i, j, l, m) = \begin{pmatrix} \alpha_A^* \\ \alpha_B^* \\ \alpha_C^* \\ \alpha_D^* \end{pmatrix} \cdot \begin{pmatrix} P_A \\ P_B \\ P_C \\ P_D \end{pmatrix} \approx \begin{pmatrix} \alpha^*(0, 1, 1, 0) \\ \alpha^*(0, 1, 0, 1) \\ \alpha^*(1, 0, 1, 0) \\ \alpha^*(0, 1, 1, 0) \end{pmatrix} \cdot \begin{pmatrix} P_{ij,M} P_{lm,NM} \\ P_{ij,M} P_{lm,M} \\ P_{ij,NM} P_{lm,NM} \\ P_{ij,NM} P_{lm,M} \end{pmatrix} = \alpha_p \cdot P. \tag{11}$$

Note that all probabilities add up to one, viz.,  $\mathbf{1} \cdot \mathbf{P} = 1$ . Further, Eq. 11 assumes that the effect of surface properties

outweighs the effect of agglomerate size and that, therefore, the collision efficiencies of primary particles  $\alpha_p$  are sufficient. This simplification is supported by theoretical and experimental studies [42] and reduces computational effort, as the numerical integration of Eq. 5 is not required for all class combinations. Furthermore, this allows for a promising integration of the PBE into a hybrid model as discussed in Sect. 2.3. Further justification arises from the fact that the resulting collision efficiencies are corrected by the aforementioned empirical factor given by Selomulya et al. [68] as shown in the following:

$$\alpha_{12} = \alpha_{12}^* \frac{\exp\left(-x_S \left(1 - \frac{r_1}{r_2}\right)\right)}{(r_1 r_2)^{y_S}} \quad (12)$$

with model parameters  $x_S$  and  $y_S$ . The key take-away and main benefit of the collision case model is that every collision efficiency between any two given agglomerates is readily calculated based on the known agglomerate volume composition and the un-corrected collision efficiencies between primary particles  $\alpha_p$ , as discussed in [55, 61]. With this, all required equations are given to solve the PBE and to obtain the discrete, time-dependent agglomerate composition distribution  $n(i, j, t)$ .

Experimental studies on MSF, however, do not yield such detailed information but only offer integral knowledge about the amount of totally separated non-magnetic material. Thus, to close the gap between PBE and experiment, the magnetic separation step needs to be modeled. Note that this is specific to the investigated MSF-process and is not required in the general case of hetero-agglomerating systems. The class-specific separation efficiency is calculated according to

$$T(i, j) = [1 + \exp(-C_1 (r^2(i, j) X_M(i, j) - C_2))]^{-1}, \quad (13)$$

and is generally dependent on the cross-sectional area of an agglomerate and its magnetic volume fraction. Both  $C_1$  and  $C_2$  are empirical parameters. Applying Eq. 13 to the entire distribution before and after agglomeration yields the overall separation efficiency of non-magnetic particles

$$A_{NM} = \frac{\sum_{i=0}^{N_S} \sum_{j=0}^{N_S} n(i, j, \text{end}) V(i, j) (1 - X_M(i, j)) T(i, j)}{\sum_{i=0}^{N_S} \sum_{j=0}^{N_S} n(i, j, 0) V(i, j) (1 - X_M(i, j))}. \quad (14)$$

The separation efficiency states what proportion of the originally present non-magnetic particles are separated.  $A_{NM} = 1$  means that all of the material was separated, while for  $A_{NM} = 0$ , no separation took place. A more in-depth look and derivation of Eq. 13 are given in the supplementary information (SI.1).

## 2.2 Black-box models

BBMs are universal non-linear function approximators [27] and map any given input space on a desired output space. In the scope of this work, various ML algorithms are used as BBM, all of which belong to the group of supervised learning algorithms. In supervised learning, the available data set is commonly split into a train and a test sub-set. The train set is used to build the ML model by learning a relationship between its input and target variables. The model quality is then evaluated by comparing its predictions with the test data. All investigated algorithms are briefly described below in terms of their structure, application to regression problems, and relevant hyper-parameters (HP). An overview of the relevant HPs is given in Table 2 and an optimization framework is discussed in Sect. 2.4. All algorithms were implemented using the Python library Scikit-learn [53] (KNeighborsRegressor, DecisionTreeRegressor, RandomForestRegressor, MLPRegressor, and SVR). It is important to note that an ever-increasing number of other ML algorithms exists in the literature and practice that may be better suited for any given dataset. The presented models should, therefore, be seen as an exemplary selection and other ML algorithms can readily be implemented in the presented framework.

### 2.2.1 k-Nearest neighbor algorithm

Applied to regression problems, the k-nearest neighbor algorithm (kNN) calculates the target value of a test sample by averaging the corresponding target value of the k-nearest train samples in feature space, which are determined by different distance metrics (norms). Distance weighting can be applied to emphasize the influence of neighbors closer to the test set sample. For a more time-efficient search for the k-nearest neighbors, a pre-organization of the data, e.g., by means of tree-based methods, can be useful for larger data sets.

### 2.2.2 Tree-based methods

Tree-based ML methods include the Classification and Regression Tree (CART) algorithm and its extension, the Random Forest (RF) algorithm. The CART algorithm builds a binary decision tree that divides the feature space of the train set into  $J$  distinct segments (leaves) based on certain decision criteria (nodes) while minimizing a particular error function such as the mean squared error. This decision tree assigns the objects or samples of the train set to the different segments. To prevent an overly complex tree and thus overfitting, the maximum complexity of the tree can be defined

before training by certain criteria (pre-pruning), such as specifying the maximum tree depth, the minimum number of samples required to be in a segment or leaf, and/or the minimum number of samples required to split a node. The principle of the RF algorithm is to generate multiple decision trees on the same train set and then average the predictions of each tree on the test set. Bootstrap aggregating (bagging) and randomly selecting a certain number of candidates from all possible features to be considered when splitting a node generate different decision trees on the same train set.

### 2.2.3 Artificial neural networks

Artificial neural networks (ANN) with their most commonly used class, the multilayer perceptron (MLP), mimic the neural structures of living organisms. An MLP consists of at least three different layers, an input, an output, and one or more hidden layers. The input variables of the data set form the input layer, which is connected to the target variables in the output layer via a certain number of artificial neurons. The output signal of each neuron results from the sum of its weighted input signals and subsequent conversion by a so-called activation function. Backpropagation is used to optimize the weights (i.e., train the model) while minimizing an error function composed of model deviation and complexity, thus creating an optimal functional relationship. This optimization is iterated multiple times, while the step size with which the weights are adjusted is determined by the learning rate. During training, high model complexity is punished via a regularization term. In the scope of this work, only a single hidden layer was used due to the small data set.

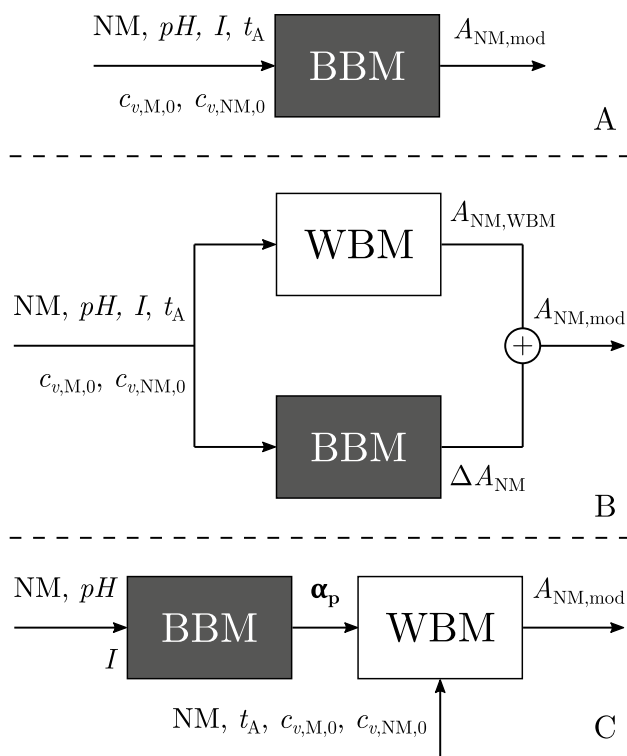
### 2.2.4 Support vector machines

Support vector machines are a statistical machine learning algorithm, and referred to as Support Vector Regression (SVR), when applied to regression tasks. In  $\epsilon$ -SVR [84], an  $\epsilon$ -insensitive region, the so-called  $\epsilon$ -tube, is introduced symmetrically around the target function to be approximated, whereby the width of the  $\epsilon$ -tube is determined by  $\epsilon$ . This region, and thus the objective function, is optimized to contain as much training data as possible within the  $\epsilon$ -tube (small prediction error) while being as flat as possible (prevention of overfitting). In this optimization problem, the regularization parameter  $C$  controls the trade-off between the flatness of the objective function and the amount up to which training data outside the  $\epsilon$ -tube, the so-called Support Vectors (SV), are punished [74]. To approximate non-linear functions, various so-called kernels are used. These transform the feature space into a higher dimensional kernel space and are composed of different kernel-internal parameters ( $\gamma$ ,  $p$ ,  $c$ ) depending on the kernel type.

## 2.3 Hybrid models

Hybrid models (HM) represent a combination of purely mechanistic models (WBM) and purely data-driven models (BBM). The hybrid approach makes it possible to combine the advantages as well as attenuate the disadvantages of its components, allowing, e.g., more accurate predictions on unseen data, improved extrapolation properties, more efficient model development, and increased transparency [87]. The WBM and BBM can be arranged in parallel or serial mode within the HM, while the serial arrangement possesses two sub-categories: (A) BBM  $\rightarrow$  WBM or (B) WBM  $\rightarrow$  BBM. Each arrangement has its individual benefits and drawbacks as well as preferred application areas. The parallel arrangement is mainly used when a WBM is available, but its accuracy is limited, e.g., due to un-modeled effects, non-linearity, or dynamic behavior [87]. In this case, the BBM compensates for the WBM's inaccuracy by correcting its predictions. There are multiple ways of combining both sub-models, but the most frequently used is pure superposition, i.e., summation of the models outputs. Arranging the HM in serial structure (B) can be applied as an alternative to the parallel arrangement, but is scarcely used in chemical and biochemical engineering [87] and therefore neglected in all further considerations. Serial arrangement (A) (further referred to as just serial arrangement), on the other hand, is frequently applied, especially in cases, where the WBM is based on conservation laws such as mass (e.g., PBE), energy, or momentum balances [87] and is therefore inherently correct. Here, the BBM represents the parts of the equations for which no reliable model exists (e.g., kinetic rates) or which is generally extracted from experimental data (e.g., material parameters).

Figure 3 shows both HM structures as well as a pure BBM applied to MSF (viz., our working example for hetero-agglomeration processes). For the case of a pure BBM, the process parameters, which will be discussed more thoroughly in Sect. 3, serve as input and the BBM directly estimates the separation efficiency  $A_{\text{NM,mod}}$ . For parallel arrangement, the BBM estimates a correction term  $\Delta A_{\text{NM}}$  that is added to the output of the WBM to yield the final result. In serial arrangement, the purpose of the BBM is to estimate the collision efficiency vector between primary particles  $\alpha_p$  (see Eq. 11) that is further processed by the WBM to calculate  $A_{\text{NM,mod}}$ . When generalizing to other hetero-agglomeration processes, it is important to remember that any data-driven model, i.e., any hybrid model, requires data to be trained on. As trivial as this might seem, it has crucial implications on the achievable information content: A pure BBM, as shown in Fig. 3 (a), will never be able to predict anything that is not directly measurable, as, e.g., the time-dependent agglomerate size and composition



**Fig. 3** Pure BBM (a), parallel HM (b), and serial HM (c) applied to MSF with corresponding input, intermediate, and output variables

distribution  $n$ . A serial HM, on the other hand, is able to estimate non-measurable variables thanks to the integrated WBM. However, any form of measurable variable (as in this case separation efficiency) is required to infer information on kinetic rates and ultimately the agglomerate population. The achievable information content of all presented models is further discussed in Sect. 3.7.

The performance of a HM is strongly dependent on the WBM's structure and accuracy: When the WBM does not include all effects (structural mismatch), parallel HMs can compensate for that and perform better. In turn, a serial connection is not expected to perform well in this case [6, 40, 87]. If the underlying WBM is accurate, however, serial arrangement is expected to perform better and further improve extrapolation properties [6, 13, 14, 82, 83, 87].

Besides the performance of the HM, the way its parameters are determined also depends on the choice of model arrangement. For the parallel model structure, parameter identification, also known as training, can be performed via direct approach. Here, priority is given to the identification of the WBM's parameters [65]. Once determined, the BBM's parameters are identified by a straightforward training with the known deviations from the experimental values (supervised learning). Training is more difficult in serial arrangement, as the necessary training data, namely the kinetic rates, are not directly measurable [52,

87]. Determination of the kinetic rates, e.g., via optimization of the WBM and subsequent parameter extraction, makes training via the direct approach possible. In this work, WBM optimization is necessary for comparing its performance to the hybrid structures, hence allowing for extraction of the kinetic rates ( $\alpha_p$ ) and application of the direct approach. Other examples for the application of this approach are found in Tholudur and Ramirez [77], Schubert et al. [67]. Parameter identification can also be performed via incremental approach, where the identification problem is decomposed into smaller parts that are solved sequentially [33]. Alternatively, the indirect or sensitivity approach adapts the backpropagation technique to the entire hybrid model. Here, the gradient of the HM's output is estimated with respect to the BBM's output and is used as error signal for parameter identification of the BBM [1, 21, 39, 52, 56]. However, this approach is often limited by the complexity of the WBM with regard to computation time in practice [1].

## 2.4 Hybrid model optimization framework: realizing a meaningful comparison

The optimal arrangement of the HM (serial or parallel) as well as the optimal ML algorithm used as BBM is highly problem-dependent and relies, e.g., on the WBM's structure, the quantity and quality of the data and the application area it will be used for. Consequently, one cannot know a priori which HM combination is best for a given problem. Therefore, it is necessary to develop a general framework that compares different ML algorithms in different HM arrangements. However, there are several requirements that have to be met to realize a meaningful and reliable comparison. Said requirements are summarized in Table 1 together with individual solution strategies. Combining all these strategies into one single framework results in a powerful tool that allows a systematic model comparison independently of application area and data quantity and quality. Note that integration is incremental in the sense that, e.g., repeated nested cross-validation (requirement 5) extends the idea of a nested cross-validation (requirement 4).

1. Supervised learning requires a split of the data set into training and test data. This is problematic, as model performance (i.e., the basis of comparison) is therefore only evaluated on a sub-set of the data. To evaluate a model on the entire data set, a  $k$ -fold cross-validation (CV) is used. Here, the data set containing  $n$  samples is divided into  $k$  different train and test sets (with  $n - n/k$  and  $n/k$  samples, respectively), whereby each sample is exactly once part of the test set. Each train set is used to train a model that is validated with the corresponding test

**Table 1** Requirements for a meaningful and reliable model comparison and corresponding solution strategies

| No | Requirement  | Solution   |
|----|--|--|
| 1  | Evaluation of model performance on the entire data set   | $k$ -fold cross-validation   |
| 2  | Comparing optimized HMs of varying complexity depending on model structure and ML algorithm        | Hyper-parameter optimization of BBM with appropriate method (Bayesian optimization) and search space (Table 2)     |
| 3  | Preventing overfitting of BBM  | Test set prediction error ( $\text{RMSE}_{\text{CV}}$ ) as optimization criterion for hyper-parameter optimization |
| 4  | Preventing overfitting due to dual use of the data set for model optimization and model evaluation | Nested cross-validation  |
| 5  | Independence of (random) divisions of the data set into train and test sets                        | Repeated nested cross-validation   |

set, thus leading to  $k$  different prediction errors. The root-mean-square error (RMSE) between experimental and calculated separation efficiency is used as measure for accuracy and defined for  $N_{\text{test}} = n/k$  test samples according to

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} (A_{\text{NM,exp}}(i) - A_{\text{NM,mod}}(i))^2} \quad (15)$$

The averaged prediction error of the cross-validation

$$\text{RMSE}_{\text{CV}} = \frac{1}{k} \sum_{i=1}^k \text{RMSE}(i) \quad (16)$$

allows for an evaluation of model quality on the entire data set.

2. To meaningfully compare HMs with respect to model structure and ML algorithm, all models have to be optimized. Otherwise, simpler HMs (with less HP) have an advantage in the sense that more complex HMs are less likely to be in an optimal state. Using the direct approach for HM training, optimization of the entire model means optimizing the HP of its BBM, as the WBM and its parameters are fixed. HP are, in contrast to model parameters (e.g., weights of the ANN), defined before training. As, again, the ideal choice of HP is problem-specific, an automated HP optimization (HPO) is required before the actual training. Depending on the model structure, training of the BBM is performed on different input and output variables (see Fig. 3). Furthermore, different ML algorithms vary in the number and type of HPs (continuous, discrete, categorical, and conditional) and hence in complexity. This results in varying optimization environments depending on the model structure and ML algorithm.

Widely used HP optimization algorithms include grid search, where the HP search space is uniformly sampled and evaluated or random search, where the HP search space is explored (semi-) randomly until a pre-defined “fitness” is reached [3, 29]. Bayesian optimization (BO) is an example

for a more sophisticated method, one of the most efficient optimization methods in terms of required iterations [8] and therefore well suited for expensive objective functions as is the case in the present study. Broadly speaking, it initially evaluates a fixed amount of randomly chosen HP combinations and builds a surrogate model based on these data that estimate the objective function. Subsequently, the BO selects the next HP combination for evaluation based on an acquisition function (e.g., the optimum of the surrogate model), essentially considering all previously acquired knowledge of the optimization problem. A commonly used surrogate model is Gaussian processes (GP) as they are easy to compute and deliver uncertainty estimates. However, GP scale poorly in higher dimensions and are unable to handle categorical and conditional HP spaces well [29]. Random forests (RF) are an alternative to GP: They allow searching over any kind of HP space (categorical, discrete, continuous, and conditional) [69, 92] and exhibit shorter computation times compared to GP, especially in higher dimensional problems [29, 69]. BO-RF does not provide an uncertainty estimate per se, although empirical methods have been shown to work well in practice [28]. Furthermore, BO-RF possesses poor extrapolation abilities and requires additional work when maximizing the acquisition function [69]. Nevertheless, BO-RF was applied in this study, as the efficiency and universal applicability of BO-RF allows for the optimization of any ML algorithm in any HM arrangement in a wide HP search space. It should be noted though that the applied surrogate model of the BO or optimization algorithm in general may easily be altered in the HPO framework to account for individual properties of the investigated data and models. The HP considered in this work with respective value ranges are shown in Table 2 for each ML algorithm. Note that the HP search spaces are data-set-specific and were pre-selected for the given problem. However, as shown in Sect. 3.8, the proposed framework allows to re-evaluate the search space at a later time. Due to the compatibility to the methods from Scikit-learn the implementation of the BO was done



**Table 2** Considered HPs with corresponding search ranges for HPO of all implemented ML algorithms

| ML algorithm                          | Hyper-parameter                        | Range  |
|---------------------------------------|--|--|
| kNN <sup>1</sup>                      | $k$ (d)                                | $\mathbb{N} \in [1, 10]$                                   |
|                                       | Norm (ca)                              | {Manhattan, Euclidean}                                     |
|                                       | Weighting (ca)                         | {Uniform, distance}  |
| CART <sup>2</sup> , RF <sup>*,2</sup> | Max. tree depth (d)                    | $\mathbb{N} \in [2, 9]$                                    |
|                                       | Min. samples per split (d)             | $\mathbb{N} \in [2, 8]$                                    |
|                                       | Min. samples per leaf (d)              | $\mathbb{N} \in [1, 8]$                                    |
|                                       | Number of trees* (d)                   | $\mathbb{N} \in [2, 70]$                                   |
| ANN <sup>1</sup>                      | Number of neurons (d)                  | $\mathbb{N} \in [3, 70]$                                   |
|                                       | Activation function (ca)               | {Linear, sigmoidal, tanh, ReLU}                            |
|                                       | Learning rate (co)                     | $\mathbb{R}^+ \in [10^{-5}, 10^{-1}]$ , $\log_{10}$ -scale |
|                                       | Regularization term (co)               | $\mathbb{R}^+ \in [10^{-7}, 10^{-1}]$ , $\log_{10}$ -scale |
|                                       | Number of iterations (d)               | $\mathbb{N} \in [10^2, 10^5]$                              |
| SVM <sup>1</sup>                      | Kernel (ca)                            | {Linear, polynomial, sigmoidal, RBF}                       |
|                                       | $\epsilon$ (co)                        | $\mathbb{R}^+ \in [10^{-4}, 10^{-1}]$ , $\log_{10}$ -scale |
|                                       | $C$ (co)                               | $\mathbb{R}^+ \in [10^{-6}, 10^6]$ , $\log_{10}$ -scale    |
|                                       | $\gamma$ (co), (cd)                    | $\mathbb{R}^+ \in [10^{-6}, 10^2]$ , $\log_{10}$ -scale    |
|                                       | $p$ (d), (cd)                          | $\mathbb{N} \in [1, 5]$                                    |
|                                       | $c$ (co), (cd)                         | $\mathbb{R}^+ \in [0, 1]$                                  |
|                                       |  |  |
| <sup>1,2</sup> Pre-processing         | Process parameter                      | Method   |
|                                       | NM                                     | {Ordinal encoding, one-hot encoding} <sup>1,2</sup>        |
|                                       | $I$                                    | {None, $\log_{10}$ + min-max normalization} <sup>1</sup>   |
|                                       | pH, $t_A$ , $c_{v,M,0}$ , $c_{v,NM,0}$ | Min-max normalization <sup>1</sup>                         |

HP types are: continuous (co), discrete (d), categorical (ca), and conditional (cd). HP that are not explicitly given or defined elsewhere are left constant on their respective default value of the used method from the Scikit-learn package. Different pre-processing methods for scaling the input parameters (features) to the value range [0, 1] were also considered in HPO

with the BayesSearchCV method from the Python library Scikit-Optimize.

3. An ML model is considered overfit, when it achieves high accuracy on train data, but low accuracy on (unseen) test data, i.e., low extrapolation ability. On the other hand, a model is underfit, when no sufficient accuracy is achieved at all. Overfitting should generally be avoided and is especially dangerous when comparing different HMs, as some structures may be more or less prone to it. Overfitting can be reduced through HPO on an appropriate search space (e.g., integration of regularization terms in ANN and SVR or maximum tree depth in CART) and by employing an appropriate optimization criterion. Here, the prediction error on the test set is suitable, since it indicates both over- and underfitting. To identify an optimal HP combination representative for the entire data set, HPO was based on  $k$ -fold CV. Therefore, the cross-validation error  $RMSE_{CV}$  (see Eq. 16) is used as optimization criterion.

4. If both HPO and model evaluation are performed on the basis of  $k$ -fold CV, dual use of the data set can lead to optimistic model performances (overfitting) [85]. The extent of overfitting is greater the smaller the data set [80]. In Shaikhina and Khovanova [70], a data set with fewer than ten samples per

feature is referred to as small. Moreover, this type of overfitting depends on the number of HP considered in the HPO [10] and thus on the ML algorithm. Since the data set used in this work is small (see Sect. 3.2) and the number of HP varies depending on the ML algorithm used (see Table 2), this type of overfitting has to be addressed. This is done using a procedure called double or nested CV. Here, the (inner)  $k$ -fold CV of the HPO is embedded into the (outer)  $k$ -fold CV of the model evaluation. HPO thus only operates on the train data of the respective fold. The concept of nested CV dates back on an early study of Stone [75] and is also described in more recent works, as, e.g., by Rendall and Reis [60].

5. Since the test/train split during  $k$ -fold CV is a stochastic process, both HPO and model evaluation will exhibit variance for repeated calculations. This variance is stronger in smaller data sets [71]. To minimize these deviations, each  $k$ -fold CV (both for HPO and model evaluation) is repeated multiple times, with different train and test sets selected in each repetition, which is referred to as repeated nested CV [18, 36].

The final framework of repeated nested CV is given in Algorithm 1 and contains all aspects described above. It therefore allows for a meaningful and fair model comparison.

**Algorithm 1** Hybrid model optimization framework with repeated nested cross validation and integrated HP optimization.

**Require:** arr  $\in$  [serial, parallel, pure BBM]

**Require:** algo  $\in$  [kNN, CART, RF, ANN, SVM]

```

1: data = ReadData(arr)
2: InitializeParameters(arr, algo)
3: for  $o \leq$  outer repetitions do
4:   for  $k \leq$  outer  $k$ -folds do
5:     train, test = SplitTrainTest(data, k)
6:     HPopt = BO(train)
7:     BBM( $o, k$ ) = TrainModel(train, HPopt)
8:     HM = CombineModels(BBM( $o, k$ ), WBM, arr)
9:     RMSE( $k$ ) = EvaluateModel(HM, test)
10:   end for
11:   RMSECV( $o$ ) = Mean(RMSE)
12: end for
13: RMSEHM = Mean(RMSECV)
14:
15: function BO(data)
16:   HP = HPinit
17:   for  $n \leq$  iterations BO do
18:     for  $i \leq$  inner repetitions do
19:       for  $k \leq$  inner  $k$ -folds do
20:         train, test = SplitTrainTest(data, k)
21:         BBM = TrainModel(train, HP)
22:         RMSE( $k$ ) = EvaluateModel(BBM, test)
23:       end for
24:       RMSECV( $i$ ) = Mean(RMSE)
25:     end for
26:     cost = Mean(RMSECV)
27:     HP = UpdateHP(cost)
28:   end for
29:   return HP
30: end function

```

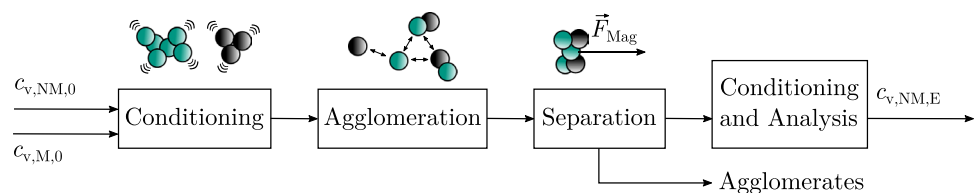
### 3 Case study: magnetic seeded filtration

#### 3.1 Experimental procedure and parameters

This study uses data from MSF experiments that are

performed similarly to Rhein et al. [63] and are shown schematically in Fig. 4 together with the relevant variables. Two individual studies (A and B) were performed with their respective parameters given in Table 3. A mono-disperse silica–magnetite composite (SiO<sub>2</sub>-MAG) was used as seed

**Fig. 4** Schematic representation of the MSF experiment and relevant variables



**Table 3** Varied experimental parameters

| Data set | M                     | NM               | $c_{v,M}$ (vol%)      | $c_{v,NM}$ (vol%)     | $t_A$ (min) |
|----------|-----------------------|------------------|-----------------------|-----------------------|-------------|
| A        | SiO <sub>2</sub> -MAG | SiO <sub>2</sub> | $2.80 \times 10^{-3}$ | $3.75 \times 10^{-3}$ | 10          |
| A        | SiO <sub>2</sub> -MAG | ZnO              | $2.80 \times 10^{-3}$ | $3.75 \times 10^{-3}$ | 10          |
| B        | SiO <sub>2</sub> -MAG | SiO <sub>2</sub> | $1.40 \times 10^{-3}$ | $1.50 \times 10^{-2}$ | 5           |
| B        | SiO <sub>2</sub> -MAG | ZnO              | $1.40 \times 10^{-3}$ | $7.14 \times 10^{-3}$ | 5           |

All other parameters are either given explicitly or kept constant throughout the study

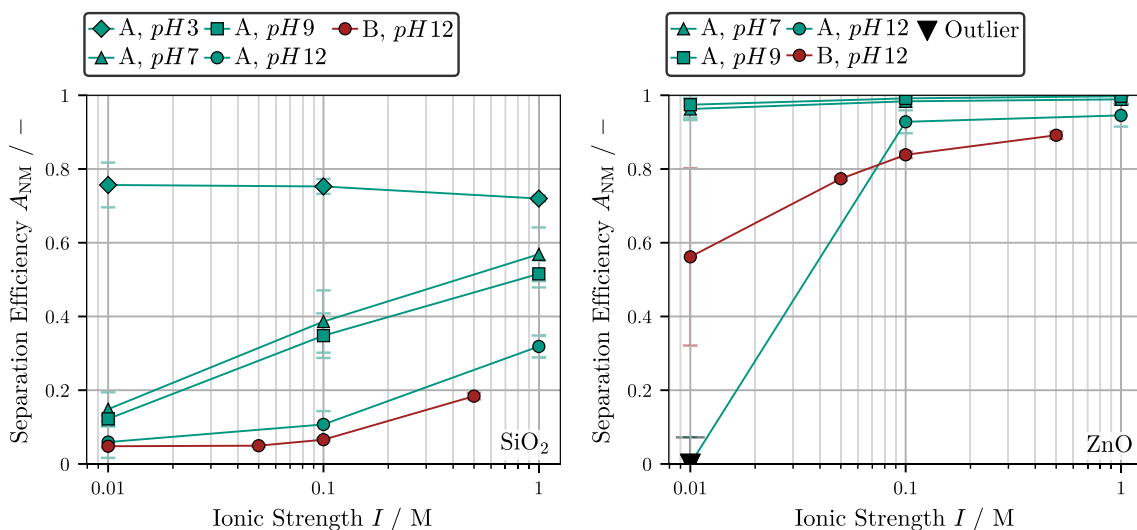
material, while the non-magnetic particle system was varied between a poly-disperse SiO<sub>2</sub> and poly-disperse ZnO powder. More information on these particle systems is given in Appendix 2. Stock suspensions of the dry particle systems were prepared by sonification with the Digital Sonifier 450 from Branson Ultrasonics Co. (conditioning). A sample of the stock suspension  $P_0$  was taken and its actual concentration determined, taking deviations in preparation into account. Subsequently, the respective volumes of the stock suspensions to achieve the desired volume concentrations  $c_{v,i}$  were transferred into a snap-on lid vial. The suspension parameters ionic strength  $I$  and pH were adjusted by addition of 2 M NaCl solution, and 0.5 M HCl or NaOH solution, respectively. Finally, the experimental volume was filled to  $V_L = 30$  mL with ultrapure water. The suspension was agitated for the agglomeration time  $t_A$  in the laboratory shaker Vortex Genius 3 from IKA GmbH. After agglomeration, a ferromagnetic separation matrix was immersed in the suspension and the vial was positioned next to a permanent magnet. Magnetic separation was performed for  $t_S = 2$  min. A representative sample  $P_E$  was taken from the non-separated suspension and sonified again achieve comparable

levels of dispersity. Samples  $P_0$  and  $P_E$  were analyzed via UV-Vis spectroscopy and the volume concentrations were determined through previously performed, substance- and system-specific calibrations. Separation success is quantified via the separation efficiency of the non-magnetic component

$$A_{NM} = 1 - \frac{c_{v,NM,E}}{c_{v,NM,0}} \quad (17)$$

### 3.2 Experimental results and data analysis

Figure 5 shows the experimentally determined separation efficiencies for SiO<sub>2</sub> (left) and ZnO (right) for various values of pH and ionic strength  $I$ . The experimental separation efficiency data is published here <https://doi.org/10.5445/IR/1000154994>. Regarding the separation of SiO<sub>2</sub>, a decrease in pH results in an increase in  $A_{NM}$ . This is explained by the zeta potentials shown in Fig. 12: At pH = 12, both SiO<sub>2</sub> and SiO<sub>2</sub>-MAG have a high, negative charge that results in a strong electrostatic repulsion, low hetero-agglomeration, and ultimately low separation efficiency. The zeta potential decreases with decreasing pH and both particle systems are even oppositely charged at pH = 3, resulting in high separation efficiencies. Generally, ZnO is separated more efficiently than SiO<sub>2</sub> for any given pH. Again, this is retraced to lower surface charge of ZnO (see Fig. 12) and therefore lower electrostatic repulsion. An increase in ionic strength leads to an increased separation efficiency for equally charged particles. This is due to a decreased DEBYE length (see Eq. 8), leading to a shorter range in the repulsive electrostatic interactions, enhanced agglomeration, and separation.



**Fig. 5** Results of the MSF experiments. Left: experiments with SiO<sub>2</sub>; right: experiments with ZnO. Two different studies (A and B, see Table 3) were performed which is indicated through coloring. The variation of the pH value is indicated by different markers

When comparing results from both studies, the separation efficiencies of A are generally slightly above the corresponding results of B. This is explained by a lower relative concentration of magnetic to non-magnetic particles in study B (see Table 3). However, the experimental datapoint for ZnO at  $\text{pH} = 12$  and  $I = 0.01 \text{ M}$  in study A is noteworthy: No separation was achieved, although the corresponding datapoint in study B showed high separation in the range of 40–60%. This is clearly an inconsistency that might be retraced to experimental error. Said datapoint is marked in Fig. 5 and henceforth referred to as outlier, as it will again be important for comparing model prediction ability in Sect. 3.5.

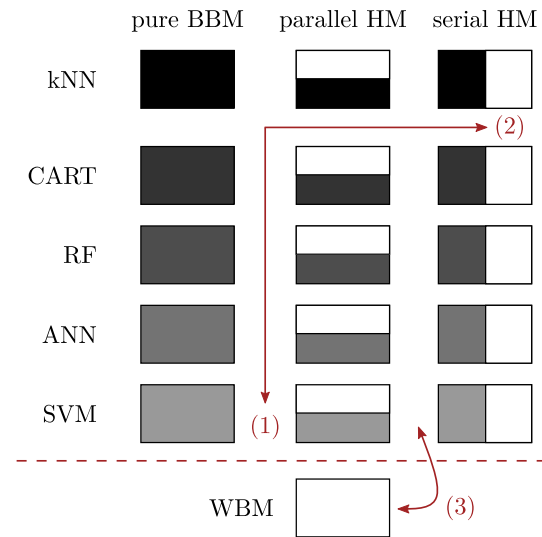
### 3.3 Performance and challenges white-box model

As stated by Praetorius et al. [55], calculating the agglomeration efficiencies  $\alpha$  ab initio remains very challenging, despite having theoretical descriptions given in Eqs. 5, 11, and 12. The main reason lies both in the availability and the accuracy of process and material-specific data. In the presented case, especially  $\bar{G}$ ,  $A_H$  and  $\zeta$  are prone to error and require adjustment to experimental data.  $\bar{G}$  is adjusted directly, while correction factors are introduced that scale  $\zeta$  and  $A_H$  according to

$$\zeta_i^* = C_\zeta \zeta_i \quad (18)$$

$$A_{H,ii}^* = C_{A,i} A_{H,ii}. \quad (19)$$

As all zeta potentials are measured identically, the error is assumed to be independent of the particle system and  $C_\zeta$  is used globally. However, HAMAKER constants depend drastically on the particle surface properties [81] and are therefore scaled with material-specific correction factors. This results in five model parameters that are optimized with algorithms of the Python toolkit SciPy with varying starting and boundary conditions. The global optimizers basinhopping [88] and shgo [17] are used to identify temporary minima in a widely scattered search field, which are further improved with the local optimizer minimize [9]. The RMSE between



**Fig. 6** Visual representation of all 16 compared model structures. Five different ML algorithms (kNN, CART, RF, ANN, and SVM) are used in two different HM arrangements (parallel and serial) and as pure BBMs. The results are further compared to the standalone WBM

experimental and calculated separation efficiency, defined in Eq. 15, is used as minimization criterion. The discretization scheme of the WBM was validated beforehand by comparison with analytical solutions, which is discussed in the supplementary information (SI.2). Furthermore, a preliminary grid study elaborated in the supplementary information (SI.3) ensured that the results are independent of chosen grid parameters for  $N_s = 12$  and  $s = 2.5$ . Table 4 lists the number of experimental data points  $N_E$  (each measured in triplicate), the optimal model parameters and the associated RMSE. Each material system of every data set is optimized individually, and additionally, all experimental data points are optimized together. The first procedure naturally yields the lowest RMSE and, therefore, best agreement between WBM and experiment. These results are used to extract the agglomeration efficiencies  $\alpha_p$  that are necessary for training the BBM in the serial HM arrangement (direct approach, see Fig. 3). Optimizing all data points at once is bound to yield

**Table 4** Results of the WBM parameter optimization with corresponding RMSE values

| Data set            | $N_E$ | $\bar{G}(\text{s}^{-1})$ | $C_\zeta$ | $C_{A,\text{SiO}_2}$ | $C_{A,\text{ZnO}}$   | $C_{A,\text{MAG}}$   | RMSE (%) |
|---------------------|-------|--------------------------|-----------|----------------------|----------------------|----------------------|----------|
| A, SiO <sub>2</sub> | 12    | 0.49                     | 0.11      | $2.4 \times 10^{-3}$ | –                    | $5.3 \times 10^{-3}$ | 9.6      |
| A, ZnO              | 9     | 2.0                      | 0.30      | –                    | $2.5 \times 10^{-2}$ | $1.3 \times 10^{-1}$ | 1.8      |
| B, SiO <sub>2</sub> | 4     | 0.51                     | 0.10      | $6.4 \times 10^{-3}$ | –                    | $1.4 \times 10^{-2}$ | 2.7      |
| B, ZnO              | 4     | 4.2                      | 0.16      | –                    | $4.1 \times 10^{-3}$ | $2.2 \times 10^{-2}$ | 0.21     |
| A + B               | 29    | 19.2                     | 0.29      | $2.1 \times 10^{-2}$ | $1.8 \times 10^{-2}$ | $9.6 \times 10^{-2}$ | 18.1     |

Each material system of every data set is optimized individually and additionally, all experimental data points are optimized together (A + B)

higher RMSE values due to the inconsistency of the underlying data set (outlier) discussed in Sect. 3.2. However, as all other models also operate on the entire data set, these WBM results are considered more appropriate for comparison and subsequently used.

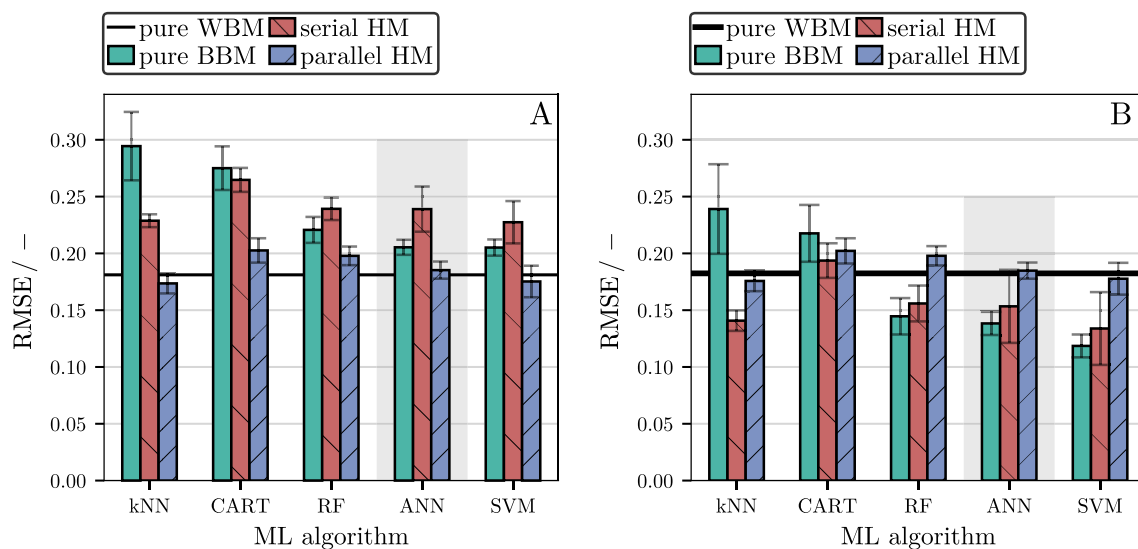
### 3.4 Study structure for model comparison

The framework from Sect. 2.4 was used to optimize and compare five different ML algorithms (kNN, CART, RF, ANN, and SVM) in two different HM arrangements (parallel and serial) and as pure BBMs. Furthermore, these models are compared to the pure WBM, optimized on the entire data set (see Sect. 3.3). This results in a comparison of 16 different models, which are visualized in Fig. 6. A model-structure-internal comparison (1) provides information about the influence of the ML algorithm on model performance, whereas a ML-algorithm-internal comparison (2) allows conclusions about the influence of model structure. Comparing different HMs with the pure WBM (3) allows the quantification of a possible model improvement by applying hybrid modeling. It should be emphasized that the results of this study are case-specific in nature. It does not aim to draw universal conclusions on model performance, but to show the potential of the methodological framework to reliably and comprehensively compare different models on any given data set.

For a comprehensive coverage of the models' advantages and disadvantages, the comparison was carried out on the basis of several criteria: The prediction quality (a) refers to the accuracy of predicting the separation efficiencies  $A_{NM}$  from the process parameters and is quantified by the

$RMSE_{HM}$ , i.e., the mean  $RMSE_{CV}$  of the outer repetitions (see Algorithm 1). The ability of the model to interpolate between the existing data points is described by the interpolation quality (b), which provides information about the applicability of the model to new, unseen data and indicates overfitting. Finally, model transparency (c) is investigated and shows whether and how reliably the hetero-agglomeration process is represented by the respective model.

Calculations were performed using  $k_{outer} = 10$  folds for the outer CV (model evaluation) and  $k_{inner} = 5$  folds for the inner CV (HPO) (see Algorithm 1). It is common to set  $k_{inner} < k_{outer}$  and this choice of  $k$ -values has been shown empirically to deliver a good balance between bias and variance (bias–variance trade-off) of the test set errors [30]. The number of used experimental datapoints for training each model in the outer  $k$ -folds is therefore given as  $N_{train,k} = (1 - k_{outer}^{-1})N_E = 26$  (see Table 4). Each  $k$ -fold CV was repeated 10 times ( $reps_{inner} = reps_{outer} = 10$ ) for further reduction of variance. Seeds were kept constant or chosen deterministically during repeated CV, to obtain reproducible results. A critical parameter is the number of iterations  $n_{iter}$  of the BO. The optimization was initially computed with  $n_{iter} = 50$  and subsequently increased to  $n_{iter} = 100$ . Since no significant improvement in prediction quality (i.e.,  $RMSE_{HM}$ ) was found between 50 and 100 iterations, regardless of the HM, the results for  $n_{iter} = 50$  are given. To provide good starting conditions for BO, the number of randomly chosen HP-starting combinations was set to 20, which proved sufficient in preliminary studies.



**Fig. 7** Mean RMSE of all implemented models: **a** including the outlier and **b** excluding the outlier in the final calculations of RMSE. All models were trained including the outlier. The models inside the gray-shaded area (ANN) are further investigated in Fig. 8

### 3.5 Comparing prediction ability

This section compares the ability of the models to predict MSF separation efficiencies from corresponding process parameters using the RMSE as measure of accuracy. By means of this metric, it is possible to compare the prediction quality on the basis of one single value, which is advantageous with regard to clarity and interpretability. Figure 7 shows the averaged RMSE of all implemented models with corresponding standard deviations from the ten outer CV repetitions (see Eq. 16). As mentioned in Sect. 3.2, one experimental data-point is considered an outlier and drastically influences RMSE values. Although all models were trained including the outlier, Fig. 7 shows two graphs, one including (a) and one excluding (b) the experimental outlier in the final calculation of the RMSE. For comparison, the RMSE of the pure WBM is drawn as a horizontal line. Due to the extensive information content of this plot and the exemplary character of this study, only the most important aspects are discussed. As stated in Sect. 3.4, model comparison is performed both in a model-structure-internal and ML-algorithm-internal way.

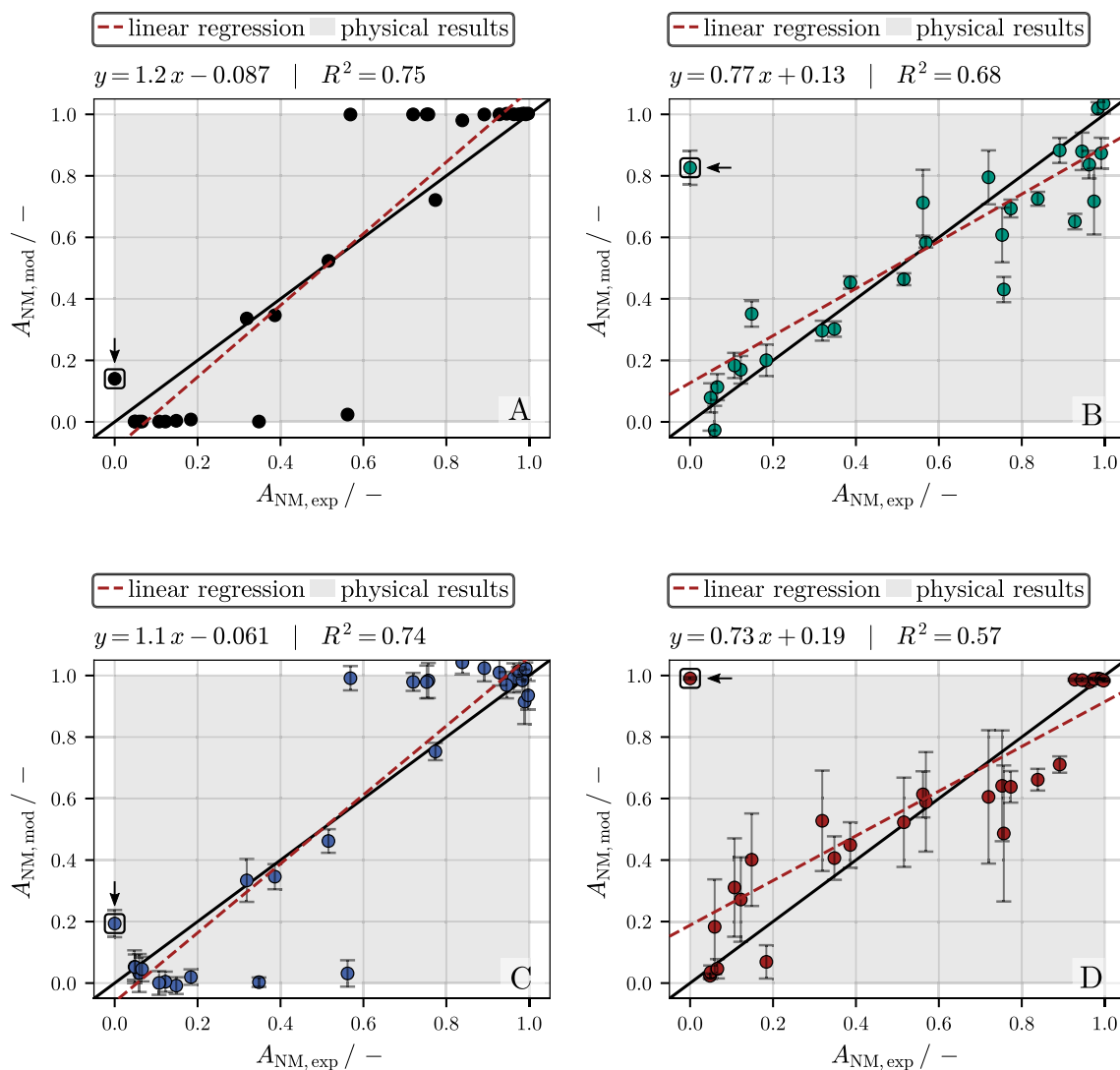
When looking at the model structure internally, it is striking that, independent of the used ML algorithm and whether outlier exclusion is performed or not, the parallel HM and the pure WBM show very similar prediction errors. Hence, the parallel HM is not able to correct the predictions of the WBM, and may even worsen the results on average (CART and RF). Since no ML algorithm leads to a significant improvement, the cause for this probably lies in insufficient data quality and quantity. In essence, the experimental data do not appear to contain a physically based connection between process parameters and deviation of the WBM [82]. Both the serial HM and pure BBM exhibit a strong dependence on outlier exclusion, which is apparent in the large decrease in RMSE for all ML algorithms when moving from Fig. 7a to 7b. This emphasizes the importance of data analysis and outlier detection prior to applying HMs, to avoid misinterpretations of prediction ability. If no outlier exclusion is performed (a), both model structures, without exception, have higher RMSE values than the pure WBM. However, if the outlier is excluded (b), serial HMs show lower prediction errors as the pure WBM for nearly all ML algorithms and pure BBMs for the more complex algorithms RF, ANN, and SVM. Extending the WBM by an appropriate BBM for taking over the error-prone part of determining the collision efficiencies, therefore, leads to a significant improvement in prediction accuracy. This can be considered a worst-case estimate, as exclusion of the outlier before training will naturally further enhance model performance. Additionally, the serial HM shows generally lower prediction errors for all ML algorithms compared to the parallel arrangement, which infers that the WBM does

not have severe un-modeled effects or structural mismatch (see Sect. 2.3). Both points indicate that the calculation of kinetic rates is indeed the limiting factor for WBM accuracy and that an appropriate BBM can attenuate this effect.

While the choice of ML algorithm has no significant impact on RMSE in serial arrangement, it is crucial for pure BBMs: kNN and CART show significantly higher prediction errors compared to the other implemented algorithms. Both algorithms predict the target variables in a similar way, i.e., by averaging target variables of training samples with similar features to predict the test samples. The small size of the data set and the associated sparse distribution of samples in the feature space therefore limit the prediction quality. As mentioned in Sect. 2.2, other ML algorithms that are not tested in this study (as, e.g., multiple linear regression and its variations) might be able to further decrease prediction errors.

It is striking that although a variety of model architectures were trained multiple times, none was able to achieve near perfect prediction accuracy. All models (including the pure WBM) appear to be offset by a minimum RMSE of about 0.1. Again, this may be attributed to inconsistencies and experimental errors in the underlying data set and emphasizes the robustness of the framework against overfitting.

As already mentioned, misinterpretations can occur when solely looking at the single RMSE-value. This is due to the significant influence of outliers on this metric. To detect these outliers, it is helpful to plot the separation efficiencies predicted by the model  $A_{NM,mod}$  against the experimentally determined values  $A_{NM,exp}$ . This enables outlier detection via a data-point-specific evaluation of prediction accuracy. Figure 8 shows these plots for the pure WBM (a), the pure BBM (b), and both parallel (c) and serial (d) hybrid model using ANN as ML algorithm. For reference, the respective models are indicated by a gray-shaded rectangle in Fig. 7. These plots are exemplary and all 16 plots for the investigated model architectures are given in the supplementary information (SI.4). These plots clearly show that the outlier is poorly predicted by both the serial HM and pure BBM (top left corner). This phenomenon is observable for all other ML algorithms and explains the large decrease in RMSE for these model structures when the outlier is excluded. Both the serial HM and pure BBM not being able to predict the outlier regardless of repeated calculations seems bizarre at first glance. However, this can be interpreted as proof for a well-functioning overfitting prevention of the methodological framework. Figure 8 further shows that the pure WBM performs poorly in the intermediate range of separation efficiency ( $0.2 < A_{NM} < 0.8$ ) and that the parallel HM is not able to compensate for this effect. Qualitatively, the performance of the serial HM and the pure BBM is better in this area, again emphasizing why only looking at RMSE is insufficient for a holistic model evaluation. Figure 8 indicates the



**Fig. 8** Predicted vs. experimentally determined separation efficiencies for **a** pure WBM, **b** pure BBM (ANN), **c** parallel HM (ANN), and **d** serial HM (ANN). The gray-shaded rectangle shows the range of

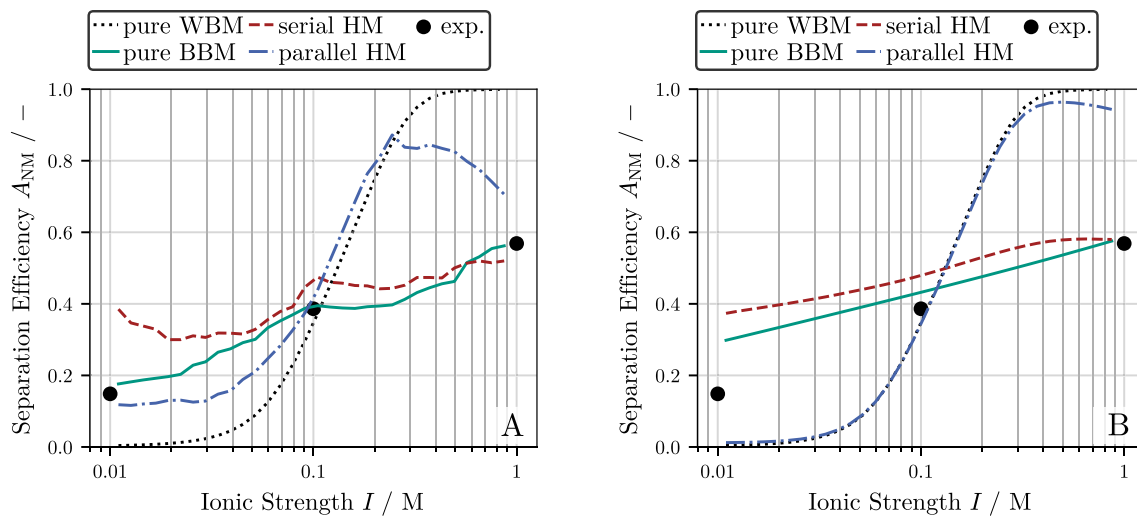
physical results ( $A_{NM} \in [0, 1]$ ), while a linear regression visualizes performance. The outlier is framed and marked with an arrow

range of physical results ( $A_{NM} \in [0, 1]$ ) by gray-shaded rectangles. Both pure WBM and serial HM are strictly bound to this area, while the pure BBM and parallel HM produce non-physical results on both ends of the spectrum. For the pure BBM with ANN as ML algorithm, this might be remedied by choosing a more appropriate activation function in the output layer. Nevertheless, this underlines the benefits of having a mechanistic and therefore physically based model produces the final results.

### 3.6 Comparing interpolation ability

This chapter investigates the models' predictive performance on new, unseen data, in between the existing data points in feature space (interpolation) to identify potential overfitting.

An overfitted model generally exhibits low training error paired with high prediction errors. Strong overfitting is especially apparent in cases where a monotonous function or relationship is mapped in a highly non-monotonous way. As can be seen in Fig. 5 and is explained in Sect. 2.1, the separation efficiency is monotonically increasing with ionic strength and therefore a good test case for overfitting. Thus, a new input data set with varying ionic strength between 0.01 M and 1 M (32 logarithmically spaced points) was generated and estimated with each previously—i.e., during the optimization procedure described in algorithm 1—trained model. All other input parameters were kept constant at their respective values from data set A ( $\text{SiO}_2$ ,  $\text{pH} = 7$ ). It should be emphasized that this only represents a small part of feature space and the subsequently discussed statements



**Fig. 9** Averaged curves of predicted separation efficiencies over ionic strength for pure WBM, pure BBM, and each HM model structure for **a** kNN and **b** ANN as ML algorithm. The experimentally determined separation efficiencies (data set A: SiO<sub>2</sub>, pH 7; see Fig. 5) are also shown

should therefore not be generalized. The resulting 100 predictions ( $k_{\text{outer}} \times \text{reps}_{\text{outer}}$ ) are averaged and shown in Fig. 9 for all model structures with kNN (a) and ANN (b) as ML algorithm. These two algorithms were chosen as examples, since ANN is a representative for a generalizing and kNN for a non-generalizing algorithm. The plots for all other ML algorithms are given in the supplementary material (SI.5).

As data set A (SiO<sub>2</sub>, pH = 7) is part of the above mentioned intermediate range in separation efficiency, Fig. 9 illustrates once more that the pure WBM has difficulties describing the real process behavior. The experimental data-points for  $I = 0.01$  M and  $I = 1$  M are not reproduced well by the WBM, although they were used during training and the WBM can therefore be considered underfit rather than overfit. As the underlying PBE and their implementation were validated beforehand (SI.2), the low performance of the WBM is either due to inaccuracy of the model equations (Eqs. 4, 5, 11, 12) or of the correction parameters (see Sect. 3.3) used for estimating the agglomeration kernels. Inaccurate model equations can be replaced or adjusted by resolved numerical simulation methods [19, 59], which, however, require additional material data and significantly increase the complexity and computation time of the PBE. Similarly, applying more detailed (viz. a larger number of) correction parameters for unknown material parameters may increase model accuracy but will also increase the complexity of both the PBE and the optimization procedure. This emphasizes a major benefit of serial HMs: The uncertainties regarding model equations and material parameters are all wrapped inside the BBM and therefore drastically reduce model complexity of the WBM, while simultaneously increase the overall prediction accuracy (see Sect. 3.5).

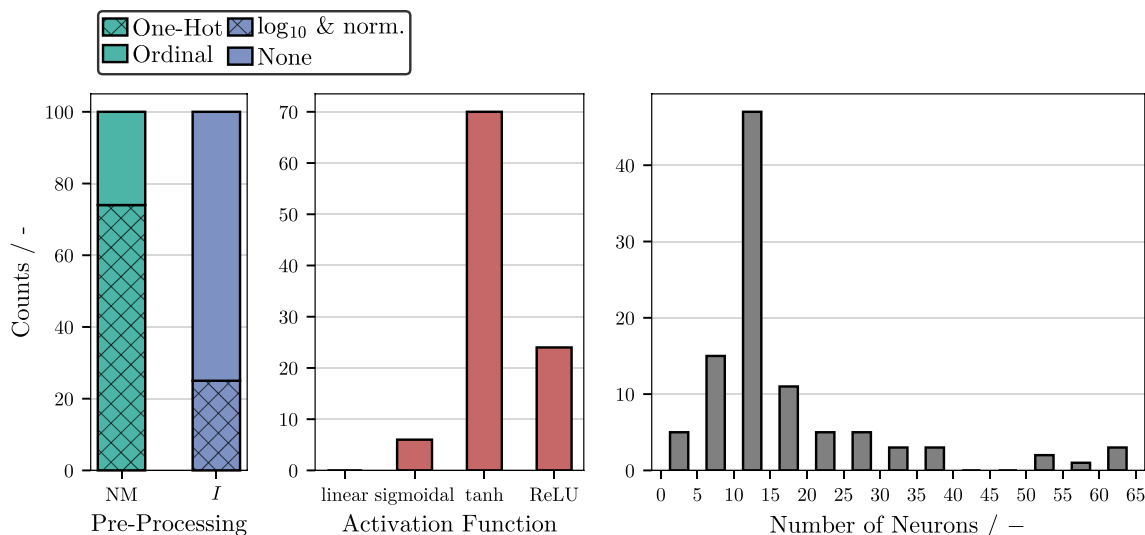
Similarly to the results discussed in Sect. 3.5, the parallel arrangement did not yield a significant improvement, neither for kNN nor ANN. Using kNN as ML algorithm (a), a correction of the separation efficiencies closer to the discrete experimental values is observed, but the representation of the dynamic behavior in between those reference points remains poor. A decrease of separation efficiency with increasing ionic strength represents non-physical behavior, which also occurs for pure BBM or serial HM with kNN as ML algorithm. Using ANN as ML algorithm (b), the interpolated curves appear smooth, which better represents the physical behavior. This smoothness is explained by the fact that generalizing ML algorithms, unlike kNN, CART, or RF, learn a functional relationship between process parameters and separation efficiency during model training.

Generally, pure BBM and serial HM are in better accordance to the expected slowly and monotonically increasing dynamic compared to pure WBM and parallel HM, thus indicating better interpolation ability. This trend is apparent for both ML algorithms, while the pure BBM is slightly closer to the experimental data points.

### 3.7 Comparing transparency

The model comparison with regard to prediction and interpolation ability showed similar performances for the pure BBM and the serial HM, when one of the more profound ML algorithms (RF, ANN, or SVM) is employed. Therefore, the question arises if all the additional effort for realizing a WBM and combining it into HMs is even justified. An argument can be made that if one is solely interested in predicting separation efficiency from a given set of process parameters, pure BBMs perform best and are both easier and faster to





**Fig. 10** Histograms of the resulting HPs for ANN in serial HM. Pre-processing of NM and *I*, the activation function, and number of neurons are considered

set up, as they do not require any physical knowledge on the problem. However, only models with incorporated WBMs provide insights into the micro-processes, i.e., give information on the time-dependent agglomerate distribution *n*. The pure BBM, being empirical in nature, does not provide any insights in this regard. Since the serial HM, generally exhibits lower prediction errors as the pure WBM as well as the parallel HM (with outlier exclusion), it can be stated that the hetero-agglomeration processes are most reliably described by this type of model structure. Even for well-performing parallel HM (e.g., on other data sets), this argument holds: As a parallel HM only provides a correction of the separation efficiency predicted by the WBM, its insight into the micro-processes, namely the calculation of the PBE, remains unchanged. This is a crucial aspect for various applications, as, e.g., model predictive control and process optimization, as one of the incentives for process modeling to begin with is to allow access to non or hard to measure phenomena.

### 3.8 A closer look on hyper-parameter optimization

Besides providing the basis for a reliable and meaningful model comparison, the methodological framework presented in Sect. 2.4 also allows for a statistical evaluation of the optimal HP combinations. By applying repeated nested CV, 100 (reps<sub>outer</sub> × *k*<sub>outer</sub>) HPOs were performed on different parts of the data set. Figure 10 shows the exemplary histograms of the resulting, optimal HP combinations for the serial HM with ANN as ML algorithm. Since it is impossible to a priori state which HP settings will provide an optimal adaption to the data set, a subsequent HP evaluation is even more decisive: This information can,

e.g., be used to narrow down the HP search space after an initially broad definition. As a result, a more precisely defined search space is expected to both reduce the number of necessary iterations and—in some cases—even produce a better optimum value.

Figure 10 shows that the number of neurons was mostly distributed in the lower range of around 10–20, and therefore, the upper limit of HP search space could be limited to, e.g., 40. Furthermore, as the linear activation function was never considered optimal, it might be removed from the HP search space. The results further indicate that one-hot encoding of the categorical input variable NM and no pre-processing of the ionic strength generally increased prediction accuracy.

## 4 Conclusions

Hybrid modeling is a promising technique for describing hetero-agglomeration processes as it is an elegant way of introducing experimental data into mechanistic models, while simultaneously attenuating major limitations of purely data-driven approaches. The presented general framework alleviates one major difficulty in applying hybrid models, namely that the optimal model architecture and ML algorithm cannot be known beforehand, as they are highly problem-dependent. By implementing a repeated nested cross validation with integrated hyper-parameter optimization, any HM combination can be compared for any given data set in a fair and objective manner to select the best-suited architecture. Additionally, the framework allows for a statistical evaluation of the selected hyper-parameters, which paves the way for further enhancement of model accuracy.

The capabilities of this approach were portrayed by a case study on magnetic seeded filtration: 16 different model architectures—including the pure WBM as well as pure BBMs, serial and parallel HMs for different ML algorithms—were compared with respect to multiple criteria. Both the serial HM and pure BBM were able to achieve a significant increase in prediction accuracy compared to the pure WBM. They also proved to enhance interpolation ability between experimental data points, especially when more elaborate ML algorithms like the ANN are used. However, as serial HMs further allow for a more accurate depiction of the ongoing micro-processes, while pure BBMs do not provide any insight, it is concluded that serial HMs are best-suited for describing hetero-agglomeration during MSF.

### Appendix 1: Geometric discretization scheme

The following section briefly summarizes the necessary equations to realize the 2-D geometric discretization scheme. For more in-depth explanations and derivations, it is referred to Kumar et al. [37, 38]. The general form of the ODE is given by

$$\frac{dn(i, j, t)}{dt} = B_{ij} + D_{ij} \quad (20)$$

The death term  $D_{ij}$  describes the loss of  $(i, j)$  due to further agglomeration with  $(l, m)$  and is calculated according to

$$D_{ij} = - \sum_{l=0}^{N_s} \sum_{m=0}^{N_s} k(i, j, l, m)n(i, j, t)n(l, m, t) \quad (21)$$

For easier notation, define  $X_i$  as partial volume of the non-magnetic component and  $Y_j$  as the partial volume of the magnetic component of agglomerate  $(i, j)$  according to

$$X_i = V_{NM}(i, j) = s^{i-1}V(1, 0) \quad | \quad Y_j = V_M(i, j) = s^{j-1}V(0, 1) \quad (22)$$

The unmodified birth term is given by

$$B_{ij}^* = \sum_{l=1}^i \sum_{a=1}^i \sum_{m=1}^j \sum_{b=1}^j \frac{1}{2} \gamma k(l, m, a, b)n(l, m, t)n(a, b, t) \quad (23)$$

$$\begin{matrix} X_{l-1} \leq X_j + X_a < X_{l+1} \\ Y_{j-1} \leq Y_m + Y_b < Y_{j+1} \end{matrix}$$

with

$$\gamma = \begin{cases} 1 & V(i, j) = V(l, m) + V(a, b) \\ \frac{1}{2} & V(i, j) \neq V(l, m) + V(a, b) \end{cases} \quad (24)$$

Note that the indices of  $\gamma(i, j, l, m, a, b)$  are not written out to improve visibility. Further, the net flux of non-magnetic volume  $x_{i,j}$  and magnetic volume  $y_{i,j}$  into cell  $(i, j)$  are defined by

$$x_{i,j} = \sum_{l=1}^i \sum_{a=1}^i \sum_{m=1}^j \sum_{b=1}^j \frac{1}{2} \gamma k(l, m, a, b)n(l, m, t)n(a, b, t)(X_l + X_a) \quad (25)$$

$$\begin{matrix} X_{l-1} \leq X_j + X_a < X_{l+1} \\ Y_{j-1} \leq Y_m + Y_b < Y_{j+1} \end{matrix}$$

$$y_{i,j} = \sum_{l=1}^i \sum_{a=1}^i \sum_{m=1}^j \sum_{b=1}^j \frac{1}{2} \gamma k(l, m, a, b)n(l, m, t)n(a, b, t)(Y_m + Y_b) \quad (26)$$

$$\begin{matrix} X_{l-1} \leq X_j + X_a < X_{l+1} \\ Y_{j-1} \leq Y_m + Y_b < Y_{j+1} \end{matrix}$$

The average properties of newly formed agglomerates in cell  $(i, j)$  are calculated as

$$\bar{X}_{i,j} = \frac{x_{i,j}}{B_{i,j}} \quad | \quad \bar{Y}_{i,j} = \frac{y_{i,j}}{B_{i,j}} \quad (27)$$

and finally, the modified birth term is calculated as

$$B_{ij} = \sum_{p=0}^1 \sum_{q=0}^1 B_{i-p,j-q}^* \lambda_{i,j}^{-,-}(X_{i-p}, Y_{j-q}) \times H[(-1)^p(X_{i-p} - \bar{X}_{i-p,j-q})]H[(-1)^q(Y_{j-q} - \bar{Y}_{i-p,j-q})]$$

$$+ \sum_{p=0}^1 \sum_{q=0}^1 B_{i-p,j+q}^* \lambda_{i,j}^{-,+}(X_{i-p}, Y_{j+q}) \times H[(-1)^p(X_{i-p} - \bar{X}_{i-p,j+q})]H[(-1)^{q+1}(Y_{j+q} - \bar{Y}_{i-p,j+q})]$$

$$+ \sum_{p=0}^1 \sum_{q=0}^1 B_{i+p,j-q}^* \lambda_{i,j}^{+,-}(X_{i+p}, Y_{j-q}) \times H[(-1)^{p+1}(X_{i+p} - \bar{X}_{i+p,j-q})]H[(-1)^q(Y_{j-q} - \bar{Y}_{i+p,j-q})]$$

$$+ \sum_{p=0}^1 \sum_{q=0}^1 B_{i+p,j+q}^* \lambda_{i,j}^{+,+}(X_{i+p}, Y_{j+q}) \times H[(-1)^{p+1}(X_{i+p} - \bar{X}_{i+p,j+q})]H[(-1)^{q+1}(Y_{j+q} - \bar{Y}_{i+p,j+q})] \quad (28)$$

with

$$\lambda_{i,j}^{\pm,\pm}(X, Y) = \frac{(X - X_{i\pm 1})(Y - Y_{j\pm 1})}{(X_i - X_{i\pm 1})(Y_j - Y_{j\pm 1})} \quad (29)$$

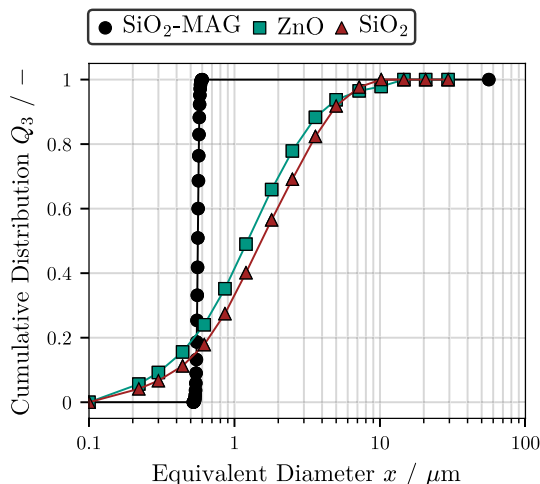
and the Heaviside function  $H$  defined as

$$H(x) = \begin{cases} 1 & x > 0 \\ \frac{1}{2} & x = 0 \\ 0 & x < 0 \end{cases} \quad (30)$$

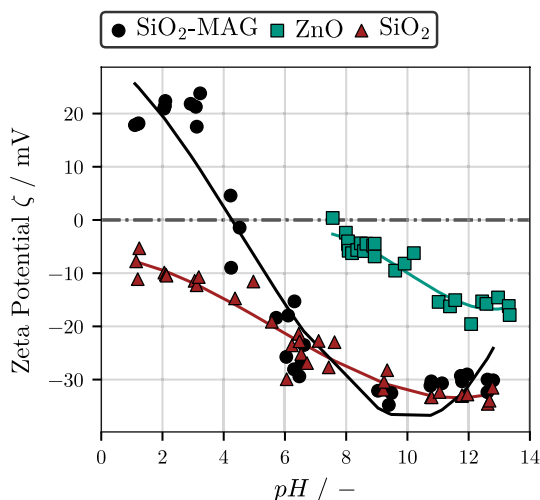
### Appendix 2: Material data

Figure 11 shows the cumulative particle-size distribution of the used particle systems. For SiO<sub>2</sub>-MAG, the distribution was measured with an analytical disc centrifuge (CPS Instruments), while ZnO and SiO<sub>2</sub> were measured by laser

diffraction (Helos, Sympatec). The pH-dependent zeta potentials are shown in Fig. 12 and were fitted with a polynomial function of degree three for continuous integration into the PBE. All zeta potentials were measured at constant ionic strength  $I = 0.1$  M and constant particle concentration  $c_v = 3.75 \times 10^{-3}$  vol%. Further information about the particle systems is given in Table 5. Zeta potential and



**Fig. 11** Cumulative particle-size distributions of the used particle systems



**Fig. 12** Zeta potentials of the used particle systems and their respective polynomial fit of third degree for usage in the PBE

**Table 5** Relevant properties of used particle systems

| Name                  | Material   | Manufacturer        | Product name                 | HAMAKER-constant $A_{iwi}$ [ $10^{-20}$ J] Bergström [4] |
|-----------------------|--|---------------------|------------------------------|--|
| SiO <sub>2</sub> -MAG | SiO <sub>2</sub> (Silica) + Fe <sub>3</sub> O <sub>4</sub> (Magnetite) | Microparticles GmbH | SiO <sub>2</sub> -MAG-AR1062 | 0.5  |
| SiO <sub>2</sub>      | SiO <sub>2</sub> (Quartz)  | Quarzwerte GmbH     | Sikron SF800                 | 1.0  |
| ZnO                   | ZnO  | Quarzwerte GmbH     | Silatherm                    | 1.9  |

size distribution data is published here <https://doi.org/10.5445/IR/1000154994>.

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1007/s00366-023-01809-8>.

**Acknowledgements** The authors acknowledge the continued funding by the DFG (German Research Foundation) in the priority program 2045 *Highly specific and multidimensional fractionation of fine particle systems with technical relevance*.

**Author Contributions** Conceptualization: FR and LH; methodology: FR and LH; formal analysis and investigation: FR and LH; writing—original draft preparation: FR and LH; writing—review and editing: FR and HN; funding acquisition: HN; resources: HN; supervision: FR and HN.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Data availability** Separation efficiency, zeta Potential and particle size distribution data is available at <https://doi.org/10.5445/IR/1000154994>.

**Declarations**

**Conflict of interest** All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

**References**

1. Azevedo C, Lee R, Portela RMC et al (2017) Hybrid ann-mechanistic models for general chemical and biochemical processes. Nova Science Publishers, Hauppauge, pp 229–256
2. Bayer B, von Stosch M, Striedner G et al (2020) Comparison of modeling methods for doe-based holistic upstream process

- characterization. *Biotechnol J* 15(5):1900,551. <https://doi.org/10.1002/biot.201900551>
3. Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. *J Mach Learn Res* 13(2)
  4. Bergström L (1997) Hamaker constants of inorganic materials. *Adv Colloid Interface Sci* 70:125–169. [https://doi.org/10.1016/S0001-8686\(97\)00003-1](https://doi.org/10.1016/S0001-8686(97)00003-1)
  5. Beykal B, Boukouvala F, Floudas CA et al (2018) Global optimization of grey-box computational systems using surrogate functions and application to highly constrained oil-field operations. *Comput Chem Eng* 114:99–110. <https://doi.org/10.1016/j.compchemeng.2018.01.005>
  6. Bhutani N, Rangaiah GP, Ray AK (2006) First-principles, data-based, and hybrid modeling and optimization of an industrial hydrocracking unit. *Ind Eng Chem Res* 45(23):7807–7816. <https://doi.org/10.1021/ie060247q>
  7. Bollas GM, Papadokonstadakis S, Michalopoulos J et al (2003) Using hybrid neural networks in scaling up an fcc model from a pilot plant to an industrial unit. *Chem Eng Process Process Intens* 42(8):697–713. [https://doi.org/10.1016/S0255-2701\(02\)00206-4](https://doi.org/10.1016/S0255-2701(02)00206-4)
  8. Brochu E, Cora VM, de Freitas N (2010) A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. <https://doi.org/10.48550/ARXIV.1012.2599>
  9. Byrd RH, Lu P, Nocedal J et al (1995) A limited memory algorithm for bound constrained optimization. *SIAM J Sci Comput* 16(5):1190–1208. <https://doi.org/10.1137/0916069>
  10. Cawley GC, Talbot NL (2010) On over-fitting in model selection and subsequent selection bias in performance evaluation. *J Mach Learn Res* 11:2079–2107
  11. Chen Y, Ierapetritou M (2020) A framework of hybrid model development with identification of plant-model mismatch. *AIChE J* 66(10):e16,996. <https://doi.org/10.1002/aic.16996>
  12. Chin CJ, Yiacoumi S, Tsouris C (1998) Shear-induced flocculation of colloidal particles in stirred tanks. *J Colloid Interface Sci* 206(2):532–545. <https://doi.org/10.1006/jcis.1998.5737>
  13. Conlin J, Peel C, Montague GA (1997) Modelling pressure drop in water treatment. *Artif Intell Eng* 11(4):393–400. [https://doi.org/10.1016/S0954-1810\(96\)00058-1](https://doi.org/10.1016/S0954-1810(96)00058-1)
  14. Corazza F, Calsavara L, Moraes F et al (2005) Determination of inhibition in the enzymatic hydrolysis of cellobiose using hybrid neural modeling. *Braz J Chem Eng* 22(1):19–29. <https://doi.org/10.1590/S0104-66322005000100003>
  15. Dosta M, Chan TT (2022) Linking process-property relationships for multicomponent agglomerates using dem-ann-pbm coupling. *Powder Technol* 398(117):156. <https://doi.org/10.1016/j.powtec.2022.117156>
  16. Elimelech M (1998) Particle deposition and aggregation : measurement, modelling and simulation, 1st edn. Colloid and surface engineering series. Butterworth-Heinemann, Woburn. <https://doi.org/10.1016/B978-0-7506-7024-1.X5000-6>
  17. Endres SC, Sandrock C, Focke WW (2018) A simplicial homology algorithm for lipschitz optimisation. *J Global Optim* 72(2):181–217. <https://doi.org/10.1007/s10898-018-0645-y>
  18. Filzmoser P, Liebmann B, Varmuza K (2009) Repeated double cross validation. *J Chemom* 23(4):160–171. <https://doi.org/10.1002/cem.1225>
  19. Frungieri G, Vanni M (2021) Aggregation and breakup of colloidal particle aggregates in shear flow: a combined monte Carlo-Stokesian dynamics approach. *Powder Technol* 388:357–370. <https://doi.org/10.1016/j.powtec.2021.04.076>
  20. Fuchs N (1934) Über die stabilität und aufladung der aerosole. *Z Phys* 89(11):736–743. <https://doi.org/10.1007/BF01341386>
  21. Georgieva P, Meireles MJ, Feyo de Azevedo S (2003) Knowledge-based hybrid modelling of a batch crystallisation when accounting for nucleation, growth and agglomeration phenomena. *Chem Eng Sci* 58(16):3699–3713. [https://doi.org/10.1016/S0009-2509\(03\)00260-4](https://doi.org/10.1016/S0009-2509(03)00260-4)
  22. Ghosh D, Moreira J, Mhaskar P (2021) Model predictive control embedding a parallel hybrid modeling strategy. *Ind Eng Chem Res* 60(6):2547–2562. <https://doi.org/10.1021/acs.iecr.0c05208>
  23. Gregory J (1975) Interaction of unequal double layers at constant charge. *J Colloid Interface Sci* 51(1):44–51. [https://doi.org/10.1016/0021-9797\(75\)90081-8](https://doi.org/10.1016/0021-9797(75)90081-8)
  24. Guo R, Liu H (2021) A hybrid mechanism- and data-driven soft sensor based on the generative adversarial network and gated recurrent unit. *IEEE Sens J* 21(22):25,901–25,911. <https://doi.org/10.1109/JSEN.2021.3117981>
  25. Hamaker HC (1937) The London-van der Waals attraction between spherical particles. *Physica* 4(10):1058–1072. [https://doi.org/10.1016/S0031-8914\(37\)80203-7](https://doi.org/10.1016/S0031-8914(37)80203-7)
  26. Honig EP, Roeberson GJ, Wiersema PH (1971) Effect of hydrodynamic interaction on the coagulation rate of hydrophobic colloids. *J Colloid Interface Sci* 36(1):97–109. [https://doi.org/10.1016/0021-9797\(71\)90245-1](https://doi.org/10.1016/0021-9797(71)90245-1)
  27. Hornik K, Stinchcombe M, White H (1989) Multilayer feed-forward networks are universal approximators. *Neural Netw* 2(5):359–366. [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
  28. Hutter F, Hoos HH, Leyton-Brown K (2011) Sequential model-based optimization for general algorithm configuration. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6683 LNCS, pp 507–523. [https://doi.org/10.1007/978-3-642-25566-3\\_40](https://doi.org/10.1007/978-3-642-25566-3_40)
  29. Hutter F, Kotthoff L, Vanschoren J (2019) Automated machine learning: methods, systems, challenges. Springer Nature, Berlin. <https://doi.org/10.1007/978-3-030-05318-5>
  30. James G, Witten D, Hastie T et al (2013) An introduction to statistical learning, vol 112. Springer, Berlin. <https://doi.org/10.1007/978-1-4614-7138-7>
  31. Jan Hagendorfer E (2021) Knowledge incorporation for machine learning in condition monitoring: A survey. In: 2021 International symposium on electrical, electronics and information engineering. Association for Computing Machinery, New York, NY, USA, ISEIEE 2021, pp 230–240. <https://doi.org/10.1145/3459104.3459144>
  32. Jeldres RI, Fawell PD, Florio BJ (2018) Population balance modelling to describe the particle aggregation process: a review. *Powder Technol* 326:190–207. <https://doi.org/10.1016/j.powtec.2017.12.033>
  33. Kahrs O, Marquardt W (2008) Incremental identification of hybrid process models. *Comput Chem Eng* 32(4):694–705. <https://doi.org/10.1016/j.compchemeng.2007.02.014>
  34. Kim BJ, Kim IK (2004) An application of hybrid least squares support vector machine to environmental process modeling. In: Liew KM, Shen H, See S et al (eds) Parallel and distributed computing: applications and technologies. Springer, Berlin, pp 184–187. [https://doi.org/10.1007/978-3-540-30501-9\\_42](https://doi.org/10.1007/978-3-540-30501-9_42)
  35. Krippel M, Dürauer A, Duerkop M (2020) Hybrid modeling of cross-flow filtration: Predicting the flux evolution and duration of ultrafiltration processes. *Sep Purif Technol* 248(117):064. <https://doi.org/10.1016/j.seppur.2020.117064>
  36. Krstajic D, Buturovic LJ, Leahy DE et al (2014) Cross-validation pitfalls when selecting and assessing regression and classification models. *J Cheminform* 6(1):1–15. <https://doi.org/10.1186/1758-2946-6-10>
  37. Kumar J, Peglow M, Warnecke G et al (2006) Improved accuracy and convergence of discretized population balance

- for aggregation: the cell average technique. *Chem Eng Sci* 61(10):3327–3342. <https://doi.org/10.1016/j.ces.2005.12.014>
38. Kumar J, Peglow M, Warnecke G et al (2008) The cell average technique for solving multi-dimensional aggregation population balance equations. *Comput Chem Eng* 32(8):1810–1830. <https://doi.org/10.1016/j.compchemeng.2007.10.001>
  39. Lauret P, Boyer H, Gatina JC (2000) Hybrid modelling of a sugar boiling process. *Control Eng Pract* 8(3):299–310. [https://doi.org/10.1016/S0967-0661\(99\)00151-3](https://doi.org/10.1016/S0967-0661(99)00151-3)
  40. Lee DS, Jeon CO, Park JM et al (2002) Hybrid neural network modeling of a full-scale industrial wastewater treatment process. *Biotechnol Bioeng* 78(6):670–682. <https://doi.org/10.1002/bit.10247>
  41. Lee DS, Vanrolleghem PA, Park JM (2005) Parallel hybrid modeling methods for a full-scale cokes wastewater treatment plant. *J Biotechnol* 115(3):317–328. <https://doi.org/10.1016/j.jbiotec.2004.09.001>
  42. Lin S, Wiesner MR (2012) Deposition of aggregated nanoparticles—a theoretical and experimental study on the effect of aggregation state on the affinity between nanoparticles and a collector surface. *Environ Sci Technol* 46(24):13,270–13,277. <https://doi.org/10.1021/es3041225>
  43. Mädler J, Richter B, Wolz DSJ et al (2022) Hybride semi-parametrische modellierung der thermooxidativen stabilisierung von pan-precursorfasern. *Chem Ing Tec*. <https://doi.org/10.1002/cite.202100072>
  44. Mayer JK, Almar L, Asylbekov E et al (2020) Influence of the carbon black dispersing process on the microstructure and performance of li-ion battery cathodes. *Energ Technol* 8(2):1900,161. <https://doi.org/10.1002/ente.201900161>
  45. McBride K, Sanchez Medina EI, Sundmacher K (2020) Hybrid semi-parametric modeling in separation processes: a review. *Chem Ing Tec* 92(7):842–855. <https://doi.org/10.1002/cite.202000025>
  46. Meneskloou P, Sinn T, Nirschl H et al (2021) Grey box modelling of decanter centrifuges by coupling a numerical process model with a neural network. *Minerals* 11(7):755. <https://doi.org/10.3390/min11070755>
  47. Meneskloou P, Sinn T, Nirschl H et al (2021) Scale-up of decanter centrifuges for the particle separation and mechanical dewatering in the minerals processing industry by means of a numerical process model. *Minerals*. <https://doi.org/10.3390/min11020229>
  48. Meng Y, Lan Q, Qin J et al (2019) Data-driven soft sensor modeling based on twin support vector regression for cane sugar crystallization. *J Food Eng* 241:159–165. <https://doi.org/10.1016/j.jfoodeng.2018.07.035>
  49. Mowbray M, Savage T, Wu C et al (2021) Machine learning for biochemical engineering: a review. *Biochem Eng J* 172(108):054. <https://doi.org/10.1016/j.bej.2021.108054>
  50. Nazemzadeh N, Malanca AA, Nielsen RF et al (2021) Integration of first-principle models and machine learning in a modeling framework: an application to flocculation. *Chem Eng Sci* 245(116):864. <https://doi.org/10.1016/j.ces.2021.116864>
  51. Nielsen RF, Nazemzadeh N, Sillesen LW et al (2020) Hybrid machine learning assisted modelling framework for particle processes. *Comput Chem Eng* 140(106):916. <https://doi.org/10.1016/j.compchemeng.2020.106916>
  52. Oliveira R (2004) Combining first principles modelling and artificial neural networks: a general framework. *Comput Chem Eng* 28(5):755–766. <https://doi.org/10.1016/j.compchemeng.2004.02.014>
  53. Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: machine learning in Python. *J Mach Learn Res* 12:2825–2830
  54. Pedrozo HA, Rodriguez Reartes SB, Chen Q et al (2020) Surrogate-model based milp for the optimal design of ethylene production from shale gas. *Comput Chem Eng* 141(107):015. <https://doi.org/10.1016/j.compchemeng.2020.107015>
  55. Praetorius A, Badetti E, Brunelli A et al (2020) Strategies for determining heteroaggregation attachment efficiencies of engineered nanoparticles in aquatic environments. *Environ Sci Nano* 7(2):351–367. <https://doi.org/10.1039/C9EN01016E>
  56. Psychogios DC, Ungar LH (1992) A hybrid neural network-first principles approach to process modeling. *AIChE J* 38(10):1499–1511. <https://doi.org/10.1002/aic.690381003>
  57. Ramkrishna D (2000) Population balances : theory and applications to particulate systems in engineering. Academic Press, San Diego. <https://doi.org/10.1016/B978-0-12-576970-9.X5000-0>
  58. Rato TJ, Delgado P, Martins C et al (2020) First principles statistical process monitoring of high-dimensional industrial microelectronics assembly processes. *Processes* 8(11):1520. <https://doi.org/10.3390/pr8111520>
  59. Reinhold A, Briesen H (2012) Numerical behavior of a multiscale aggregation model-coupling population balances and discrete element models. *Chem Eng Sci* 70:165–175. <https://doi.org/10.1016/j.ces.2011.06.041>
  60. Rendall R, Reis MS (2018) Which regression method to use? making informed decisions in data-rich/knowledge poor scenarios—the predictive analytics comparison framework (pac). *Chemom Intell Lab Syst* 181:52–63. <https://doi.org/10.1016/j.chemolab.2018.08.004>
  61. Rhein F, Ruß F, Nirschl H (2019) Collision case model for population balance equations in agglomerating heterogeneous colloidal systems: theory and experiment. *Colloids Surf A* 572:67–78. <https://doi.org/10.1016/j.colsurfa.2019.03.089>
  62. Rhein F, Scholl F, Nirschl H (2019) Magnetic seeded filtration for the separation of fine polymer particles from dilute suspensions: microplastics. *Chem Eng Sci* 207:1278–1287. <https://doi.org/10.1016/j.ces.2019.07.052>
  63. Rhein F, Schmid E, Esquivel FB et al (2020) Opportunities and challenges of magnetic seeded filtration in multidimensional fractionation. *Chem Ing Tec* 92(3):266–274. <https://doi.org/10.1002/cite.201900104>
  64. Rhein F, Kaiser S, Rhein M et al (2021) Agglomerate processing and recycling options in magnetic seeded filtration. *Chem Eng Sci* 238(116):577. <https://doi.org/10.1016/j.ces.2021.116577>
  65. Sansana J, Joswiak MN, Castillo I et al (2021) Recent trends on hybrid modeling for industry 4.0. *Comput Chem Eng* 151:107,365. <https://doi.org/10.1016/j.compchemeng.2021.107365>
  66. Santos JEW, Trierweiler JO, Farenzena M (2021) Model update based on transient measurements for model predictive control and hybrid real-time optimization. *Ind Eng Chem Res* 60(7):3056–3065. <https://doi.org/10.1021/acs.iecr.1c00212>
  67. Schubert J, Simutis R, Dors M et al (1994) Bioprocess optimization and control: application of hybrid modelling. *J Biotechnol* 35(1):51–68. [https://doi.org/10.1016/0168-1656\(94\)90189-9](https://doi.org/10.1016/0168-1656(94)90189-9)
  68. Selomulya C, Bushell G, Amal R et al (2003) Understanding the role of restructuring in flocculation: the application of a population balance model. *Chem Eng Sci* 58(2):327–338. [https://doi.org/10.1016/S0009-2509\(02\)00523-7](https://doi.org/10.1016/S0009-2509(02)00523-7)
  69. Shahriari B, Swersky K, Wang Z et al (2016) Taking the human out of the loop: a review of Bayesian optimization. *Proc IEEE* 104(1):148–175. <https://doi.org/10.1109/JPROC.2015.2494218>
  70. Shaikhina T, Khovanova NA (2017) Handling limited datasets with neural networks in medical applications: a small-data approach. *Artif Intell Med* 75:51–63. <https://doi.org/10.1016/j.artmed.2016.12.003>

71. Shaikhina T, Lowe D, Daga S et al (2015) Machine learning for predictive modelling based on small data in biomedical engineering. *IFAC-PapersOnLine* 28(20):469–474. <https://doi.org/10.1016/j.ifacol.2015.10.185>
72. Sharma N, Liu YA (2022) A hybrid science-guided machine learning approach for modeling chemical processes: a review. *AIChe J*. <https://doi.org/10.1002/aic.17609>
73. Simon LL, Fischer U, Hungerbühler K (2006) Modeling of a three-phase industrial batch reactor using a hybrid first-principles neural-network model. *Ind Eng Chem Res* 45(21):7336–7343. <https://doi.org/10.1021/ie060181z>
74. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. *Stat Comput* 14(3):199–222. <https://doi.org/10.1023/B:STCO.0000035301.49549.88>
75. Stone M (1974) Cross-validatory choice and assessment of statistical predictions. *J R Stat Soc Ser B (Methodol)* 36(2):111–133. <https://doi.org/10.1111/j.2517-6161.1974.tb00994.x>
76. Taboada-Serrano P, Chin CJ, Yiacoumi S et al (2005) Modeling aggregation of colloidal particles. *Curr Opin Colloid Interface Sci* 10(3):123–132. <https://doi.org/10.1016/j.cocis.2005.07.003>
77. Tholudur A, Ramirez WF (1996) Optimization of fed-batch bioreactors using neural network parameter function models. *Biotechnol Prog* 12(3):302–309. <https://doi.org/10.1021/bp960012h>
78. Thompson ML, Kramer MA (1994) Modeling chemical processes using prior knowledge and neural networks. *AIChe J* 40(8):1328–1340. <https://doi.org/10.1002/aic.690400806>
79. Thon C, Finke B, Kwade A et al (2021) Artificial intelligence in process engineering. *Adv Intell Syst* 3(6):2000,261. <https://doi.org/10.1002/aisy.202000261>
80. Vabalas A, Gowen E, Poliakoff E et al (2019) Machine learning algorithm validation with a limited sample size. *PLoS ONE* 14(11):1–20. <https://doi.org/10.1371/journal.pone.0224365>
81. Valmacco V, Elzbiaciak-Wodka M, Besnard C et al (2016) Dispersion forces acting between silica particles across water: influence of nanoscale roughness. *Nanoscale Horiz* 1(4):325–330. <https://doi.org/10.1039/C6NH00070C>
82. Van Can HJL, Hellinga C, Luyben KCAM et al (1996) Strategy for dynamic process modeling based on neural networks in macroscopic balances. *AIChe J* 42(12):3403–3418. <https://doi.org/10.1002/aic.690421211>
83. Van Can HJL, Te Braake HAB, Dubbelman S et al (1998) Understanding and applying the extrapolation properties of serial gray-box models. *AIChe J* 44(5):1071–1089. <https://doi.org/10.1002/aic.690440507>
84. Vapnik VN (2000) *The nature of statistical learning theory*, 2nd edn. Springer, New York. <https://doi.org/10.1007/978-1-4757-3264-1>
85. Varma S, Simon R (2006) Bias in error estimation when using cross-validation for model selection. *BMC Bioinform* 7:1–8. <https://doi.org/10.1186/1471-2105-7-91>
86. Venkatasubramanian V (2019) The promise of artificial intelligence in chemical engineering: is it here, finally? *AIChe J* 65(2):466–478. <https://doi.org/10.1002/aic.16489>
87. von Stosch M, Oliveira R, Peres J et al (2014) Hybrid semi-parametric modeling in process systems engineering: past, present and future. *Comput Chem Eng* 60:86–101. <https://doi.org/10.1016/j.compchemeng.2013.08.008>
88. Wales DJ, Doye JPK (1997) Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *J Phys Chem A* 101(28):5111–5116. <https://doi.org/10.1021/jp970984n>
89. Wang X, Chen J, Liu C et al (2010) Hybrid modeling of penicillin fermentation process based on least square support vector machine. *Chem Eng Res Des* 88(4):415–420. <https://doi.org/10.1016/j.cherd.2009.08.010>
90. Winkler M, Gleiss M, Nirschl H (2021) Soft sensor development for real-time process monitoring of multidimensional fractionation in tubular centrifuges. *Nanomaterials*. <https://doi.org/10.3390/nano11051114>
91. Wu Z, Rincon D, Christofides PD (2020) Process structure-based recurrent neural network modeling for model predictive control of nonlinear processes. *J Process Control* 89:74–84. <https://doi.org/10.1016/j.jprocont.2020.03.013>
92. Yang L, Shami A (2020) On hyperparameter optimization of machine learning algorithms: theory and practice. *Neurocomputing* 415:295–316. <https://doi.org/10.1016/j.neucom.2020.07.061>. [arXiv:2007.15745](https://arxiv.org/abs/2007.15745)
93. Zendejboudi S, Rezaei N, Lohi A (2018) Applications of hybrid models in chemical, petroleum, and energy systems: a systematic review. *Appl Energy* 228:2539–2566. <https://doi.org/10.1016/j.apenergy.2018.06.051>
94. Zhao L, You F (2019) A data-driven approach for industrial utility systems optimization under uncertainty. *Energy* 182:559–569. <https://doi.org/10.1016/j.energy.2019.06.086>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.