

Hybrid modelling of a batch separation process

Ulderico Di Caprio^a, Min Wu^a, Furkan Elmaz^b, Yentl Wouters^c, Niels Vandervorst^c, Ali Anwar^b, Siegfried Mercelis^b, Steffen Waldherr^d, Peter Hellinckx^e, M. Enis Leblebici^{a,*}

^aCenter for Industrial Process Technology, Department of Chemical Engineering, KU Leuven, Agoralaan Building B, 3590 Diepenbeek, Belgium

^bIDLab, Faculty of Applied Engineering, University of Antwerp-imec, Sint-Pietersvliet 7, 2000 Antwerp, Belgium

^cJanssen Pharmaceutica N.V., Chemical Process R&D, Turnhoutseweg 30, B-2340 Beerse, Belgium

^dKU Leuven, Department of Chemical Engineering, Celestijnenlaan 200F-bus 2424, Leuven 3001, Belgium

^eFaculty of Applied Engineering, University of Antwerp, Groenenborgerlaan 171, 2000 Antwerp, Belgium

1. Material and methods

1.1 The process

The reboiler is equipped with a jacket to supply the thermal energy needed to drive the process. The evaporated solvent leaving the reboiler is condensed and collected in the receiver tank. The operation is controlled by: 1) the jacket temperature, 2) the speed of the stirrer, 3) the flow-rate of the Ac to the reboiler from the feed tank and 4) the direction of the condensed vapor (it can either be refluxed to the reboiler or conveyed to the receiver tank where all the evaporated solvent are collected).

The *constant volume* phase is performed using the jacket to supply the heat required for the evaporation task and the fresh solvent flowrate is controlled to keep the volume inside the reboiler constant.

The process takes place at 101.5 kPa. Inert gas and a vent system are used to control the pressure within the reboiler.

1.2 The available data

The process variables used to develop the hybrid model are the following:

- 1) *Temperature of the liquid inside the reboiler*; the error of this measurement is +/- 1.5 °C.
- 2) *Temperature of the service-fluid inside the jacket*; it is the temperature at the inlet of the jacket coils. The error of this measurement is +/- 1.5 °C.
- 3) *Stirring speed* of the stirrer inside the reboiler.
- 4) *Volume of the liquid inside the reboiler*; it is measured by a radar sensor. The measurements have an uncertainty related to the stirring effect.
- 5) *Volume of the liquid inside the receiver tank*; measured by a radar sensor. This measurement bears an underestimation of the evaporated amount because of the loss on the vent gas. This bias has been estimated to be 10% of the volume inside the receiver tank.
- 6) *Mass of solvent contained in the feed tank*.
- 7) *Reflux direction*: direction of the stream that leaves the condenser. It can be either routed to the receiver tank or back to the reboiler

To compute the Ac flowrate the variable “Mass of solvent contained in the feed tank” was filtered by using a low-pass filter in order to reduce the measurement noise and its accumulation when the derivative is computed.

s1.3 The white-box model

The model was deployed in Python 3.7.4 with packages managed by using anaconda environments. The model considers the presence of three components in the liquid phase and four components in the vapor phase. (1s) and (2s) represent the mass balances for the organic solvent and for the inert gas respectively.

$$\frac{dM_i}{dt} = F_{in} * z_i^F + R * F_R(t - dt) * z_i^R(t - dt) - F_{out} * y_i \quad (1s)$$

$$\frac{dM_I}{dt} = F_{gas} - F_{out} * y_I \quad (2s)$$

In (1s), $i = \text{Ac, DCM and MeOH}$. $M_i [\text{mol}]$ is the total amount of moles within the reboiler of i , $F_{in} [\text{mol} * \text{s}^{-1}]$ is the flowrate of the fresh solvent to the reboiler, $z_i^F [\text{mol} * \text{mol}^{-1}]$ is the concentration of i in the fresh solvent stream, $R [-]$ is the reflux direction, $F_R(t - dt) [\text{mol} * \text{s}^{-1}]$ is the flowrate of condensate amount of solvent leaving the system, $z_i^R [\text{mol} * \text{mol}^{-1}]$ is the concentration of the component i within the condensate amount of solvent leaving the system, $F_{out} [\text{mol} * \text{mol}^{-1}]$ is the flowrate of the vapor leaving the reboiler, $y_i [\text{mol} * \text{mol}^{-1}]$ is the concentration of the component i in the vapor leaving the reboiler.

In (2s), $M_I [\text{mol}]$ is the total amount of inert gas within the system, $F_{gas} [\text{mol} * \text{s}^{-1}]$ is the flowrate of inert gas sent to the reboiler, $F_{out} [\text{mol} * \text{s}^{-1}]$ is the flowrate of the vapor leaving the reboiler, y_I is the concentration of the inert gas in the vapor leaving the system.

The vapor-liquid equilibrium was modelled by using the modified Raoult's law (3s).

$$P * y_i = P_i^0(T) * \gamma_i(x, T) * x_i \quad (3s)$$

where $P [\text{kPa}]$ is the pressure of the system, y_i is the vapor molar fraction of the i -th component, $P_i^0(T) [\text{kPa}]$ is the vapor pressure of the i -th component at the temperature T , $\gamma_i(x, T)$ is the activity coefficient of the i -th component within a mixture with x molar fractions at the temperature T and x_i is the liquid molar fraction of the i -th component.

The DCM-Methanol and DCM-Ac mixtures have a non-ideal liquid behavior (Khurma et al., 1983). To model the deviation from the ideality of the liquid mixtures, the non-random two liquid (NRTL) model was used (Renon and Prausnitz, 1968). The parameter required by the NRTL model were obtained by Aspen HYSYS v10. The parameter were tested using the experimental studies available in literature (Amer et al., 1956; Khurma et al., 1983; Martin et al., 1991; Nath and Prakash Dixit, 1990). Determination coefficients (R^2) above 0.97 were obtained for all the comparisons.

s1.4 The structures of the black-box models

s1.4.1 Differential evolution

To initialize the DE algorithm a population is located within the search space and the fitness function for each particle is calculated.

For each member of the population ϕ_{parent} , the following three steps are executed

1. Mutation: four different vectors $\phi_1, \phi_2, \phi_3, \phi_4$ are randomly chosen among the population. The four vectors are linear combined to obtain a mutated vector

$$\phi_{trial} = \phi_{parent} + F * ((\phi_1 - \phi_2) + (\phi_3 - \phi_4)) \quad (4s)$$

2. Crossover: the components of the trial and parent vector are shuffled to create a new trial vector $\phi_{offspring}$ using a crossover probability factor Cr whose value can vary in the range [0,1]

$$\phi_{offspring,i} = \begin{cases} \phi_{trial,i} & \text{if } rand_i \leq Cr \\ \phi_{parent,i} & \text{if } rand_i > Cr \end{cases} \quad (5s)$$

3. Selection: The fitness function $C_{fitness}(\phi)$ is calculated for the vector $\phi_{offspring}$. Its value is compared to the one generated from ϕ_{parent} . If $C_{fitness}(\phi_{offspring}) < C_{fitness}(\phi_{parent})$

than the $\phi_{offspring}$ is selected for the next generation in place of ϕ_{parent} , otherwise ϕ_{parent} is kept for the next generation.

All the steps listed above are repeated for each particle until the termination criterion is satisfied. The model training was run by using the implementation of the DE algorithm available in SciPy (Virtanen et al., 2020).

S1.4.2 Particle swarm optimization (PSO)

To initialize the PSO algorithm a population is located within the search space by giving to each particle an initial position ϕ_i and an initial random velocity v_i . For each particle the fitness score is calculated. The one with the lower fitness function is selected to be the best particle ϕ_{best}^{global} . For each particle an individual best $\phi_{best}^{individual,i}$ is assigned; during the initialization phase, the individual best corresponds to the initial position of each particle. The loop is composed by two steps repeated for each particle:

1. Velocity update: the following relation is applied to compute the velocity of each particle

$$v_{i+1} = \omega * v_i + c_p * r_p * (\phi_{best}^{individual,i} - \phi_i) + c_g * r_g * (\phi_{best}^{global} - \phi_i) \quad (6s)$$

where r_p and r_g are two vectors with the same dimension of the parameter space which components are randomly chosen in the range [0,1]. ω is the inertial parameter, c_p is the cognitive parameter and c_g is the social parameter.

2. Update the position of each particle

$$X_{i+1} = X_i + v_{i+1} \quad (7s)$$

where X_{i+1} is the position of the particle at the iteration i+1, X_i is the position of the particle at the iteration i and v_{i+1} is the updated velocity calculated through (6s).

The fitness function is calculated to the new position of the particle i-th. If the new solution has a lower fitness score than the individual best it become the new individual best for the i-th particle. If the solution has a lower fitness score than the global best it become the new global best.

All the steps listed above are repeated for each particle until the termination criterion is satisfied. The model training was run by using the implementation of the PSO algorithm available in PySwarms (Miranda, 2018).

s1.5 Fitness function

In the following list the description of the components used for the fitness function are reported:

1. *Reboiler temperature*: the mean squared error was used as benchmark

$$C_{TempReb}(\phi) = \frac{\sum_{i=1}^N (T_{reboiler}^{model}(t_i) - T_{reboiler}^{exp}(t_i))^2}{N} \quad (8s)$$

where $T_{reboiler}^{model}(t_i)$ is the value of the temperature within the reboiler predicted by the model at the time t_i , $T_{reboiler}^{exp}(t_i)$ is the experimental value of the temperature within the reboiler at the time t_i and N is the amount of sample points used for the comparison.

2. *Receiver tank volume*: the benchmark of this variable is non-trivial due to the loss in the vent gas. The loss was estimated to be up to 10% of the amount inside the receiver tank

$$C_{RecTank}(\phi) = \begin{cases} \frac{\sum_{i=1}^N |V_{RecTank}^{model}(t_i) - V_{RecTank}^{exp}(t_i)|^{1.5}}{N} & \text{if } V_{RecTank}^{exp}(t_i) < V_{RecTank}^{model}(t_i) < V_{RecTank}^{exp}(t_i) + 10\% \\ \frac{\sum_{i=1}^N (V_{RecTank}^{model}(t_i) - V_{RecTank}^{exp}(t_i))^2}{N} & \text{if } V_{RecTank}^{model}(t_i) > V_{RecTank}^{exp}(t_i) + 10\% \\ \frac{\sum_{i=1}^N (V_{RecTank}^{model}(t_i) - V_{RecTank}^{exp}(t_i))^2}{N} & \text{if } V_{RecTank}^{model}(t_i) < V_{RecTank}^{exp}(t_i) \end{cases} \quad (9s)$$

where $V_{RecTank}^{model}(t_i)$ is the value of the volume within the receiver tank predicted by the model at the time t_i , $V_{RecTank}^{exp}(t_i)$ is the experimental value of the volume within the receiver tank at the time t_i and N is the amount of sample points used for the comparison. This part of the fitness function aims to let the receiver tank profile be in the accepted error bounds of the variable. If the value of the predicted variable is within the error range, the error is less than squared, on the contrary the error is squared.

3. *Process time*: the process time predicted by the model is compared to the experimental one by accounting the absolute error.

$$C_{Time}(\phi) = t_f^{exp} - t_f^{model} \quad (10s)$$

where t_f^{exp} is the experimental end time of the process and t_f^{model} is the modelled end time of the process. This part of the fitness function is needed to avoid premature interruption of the process due to a high evaporation rate. In case the computation of the differential evolution fails because of the instabilities due to the black-box function, the cost function value is set to be 10^{14} .

s2. Results and discussion

s2.1 Parallel training

s2.1.1 PSO/MRF/Parallel

The PSO/MRF/Parallel hybrid models have a high accuracy and robustness on the training set because of the low variance and median value in Figure 3a. However, on the test set the accuracy is lower with a significant amount of runs that stopped the calculation because of mathematical instabilities (Figure 3b). The black-box parameter identified by the PSO/MRF/Parallel do not follow the reference, are outside the physical validity range and show numerical instabilities related to the presence of the denominator (Figure 4 **Error! Reference source not found.** and Figure 5).

s2.1.2 PSO/poly/Parallel

The PSO/poly/Parallel hybrid models have an high accuracy both on the training set (Figure 3a) and on the test set (Figure 3b) where no outlier can be detected. The linearity in the parameter search permits high black-box parameter identification capabilities. Most of the black-box models follow the trend of the reference and the k_{UA}^{HM} values are in the boundaries proposed in literature (Figure 4 and Figure 5

s2.1.3 DE/MRF/Parallel

Th hybrid models DE/MRF/Parallel have an high accuracy and search robustness on the training set (Figure 3a). However, on the test set it shows low generalization capabilities and mathematical instabilities (Figure 3b). The black box model profiles show instabilities in the training range, poor accuracy in following the trend of the reference and low robustness with the initial condition (Figure 4 and Figure 5).

s2.1.4 DE/poly/Parallel

The hybrid models DE/poly/Parallel show the lower fitness function values among all the model trained with the parallel approach both on the train set (Figure 3a) and on the test set (Figure 3b). This training technique returns no instabilities and high robustness to the initial population. These level of accuracy and robustness are related to the augmented prediction capabilities of the black-box functions. In this case all the values predicted by the black-box models follows the reference trend and the k_{UA}^{HM} values are in the boundaries proposed in literature (Figure 4 **Error! Reference source not found.** and Figure 5).

s2.2 Serial training

s2.2.1 PSO/MRF/Serial

The PSO/MRF/Serial training technique shows high fitness scores when evaluated on the training set. In addition, the prediction returned from the hybrid model are highly scattered (Figure 3). Despite that, on the test set the model has the higher accuracy among all the training techniques with a low variance (Figure 3b). In addition, the simulation does not experience any mathematical instabilities. The majority of the h_L^{HM} value follow the trend of the reference, even if their absolute values is lower than the reference. The value of k_{UA}^{HM} are in the boundaries suggested from the literature and follow the trend of the reference (Figure 6 and Figure 7).

s2.2.2 PSO/poly/Serial

The PSO/poly/Serial training technique return low median and variance fitness function scores both on the training set and on the test set (Figure 3a and Figure 3b). The robustness and accuracy of the hybrid models is possible thanks to the fine prediction of the function for their black-box sub-models. All the curves predicted by the PSO/poly/Serial return curves with the trend that match the reference; the values of h_L^{HM} are comparable with the reference and the values of k_{UA}^{HM} are within the range proposed by the literature (Figure 6 and Figure 7).

s2.2.3 DE/MRF/Serial

The hybrid models obtained with the DE/MRF/Serial show high prediction accuracy and robustness. The fitness function scores have a median and a variance comparable to the other models obtained with the serial approach. In addition, the calculation did not show any instabilities on the test set. The accuracy of the hybrid model is consequence of the high identification accuracy and robustness of the black-box parameters. The majority of the curves follow the trend both for the values of h_L^{HM} and k_{UA}^{HM} where the values are in the range identified from the literature (Figure 6 and Figure 7).

s2.2.4 DE/poly/Serial

The hybrid models obtained with the DE/poly/Serial are the most accurate and robust obtained in this work. The accuracy of the predictions obtained from this model is comparable with the other techniques performed with the serial training, however the variance of the fitness score calculated on the training and on the test set is the lowest experience in this study (Figure 3a and Figure 3b). The high accuracy and robustness are related to the augmented robustness in the parameter identification. The influence of the initial condition on the predicted profile is neglectable and the values of k_{UA}^{HM} are within the range identified from the literature (Figure 6 and Figure 7).

s2.2.4 Comparison of the black-box model parameter prediction in the parallel training

Further information can be deducted from Table 3. DE/poly/Parallel returns lower IoD values compared to the PSO/poly/Parallel. However, the grater capabilities of the DE are only partially respected when the MRF are used as black-box structure. The DE/MRF/Parallel shows a still lower IoD value than the PSO/MRF/Parallel when predicting k_{UA}^{HM} . On the contrary, when investigating the h_{dev}^{HM} model the PSO/MRF/Parallel obtain lower IoD than the DE/MRF/Parallel. However, the results obtained by the PSO/MRF/Parallel do not have physical sense. Thus, the lower IoD values are related

to the overfitting problem and the capability of the PSO to exploit the minimum that has already found rather than find new minimum. This is more critical on the search of the parameter of h_{dev}^{HM} rather than k_{UA}^{HM} since the function of h_L^{HM} have two independent variables while k_{UA}^{HM} has only one independent variable.

s2.3 Comparison between the two training approaches

All the techniques show the increased accuracy when trained in serial but the PSO/MRF when is evaluated on the training set. In this case, parallel training achieves a lower median fitness function value and variance on the training set. However, the prediction capabilities are not conserved when moving to the test set. Here, the parallel training present instabilities, a higher median of the fitness function together with a higher variance than the serial training. For these reasons, the PSO/MRF/Parallel overfits on the training data and returns poor prediction capabilities. On the contrary, the PSO/MRF/Serial extend its prediction capabilities also on the test set.

s2.4 Physical simulation results and model extrapolation capabilities

Figure 1s and Table 1s accent the poor prediction accuracy obtained by the PSO/MRF/Parallel and DE/MRF/Parallel trainings. The temperature inside the reboiler deviates from the experimental data at the half of the first *evaporation* phase (see Section 2.1). However, the prediction returned from all the other techniques follow the experimental trend.

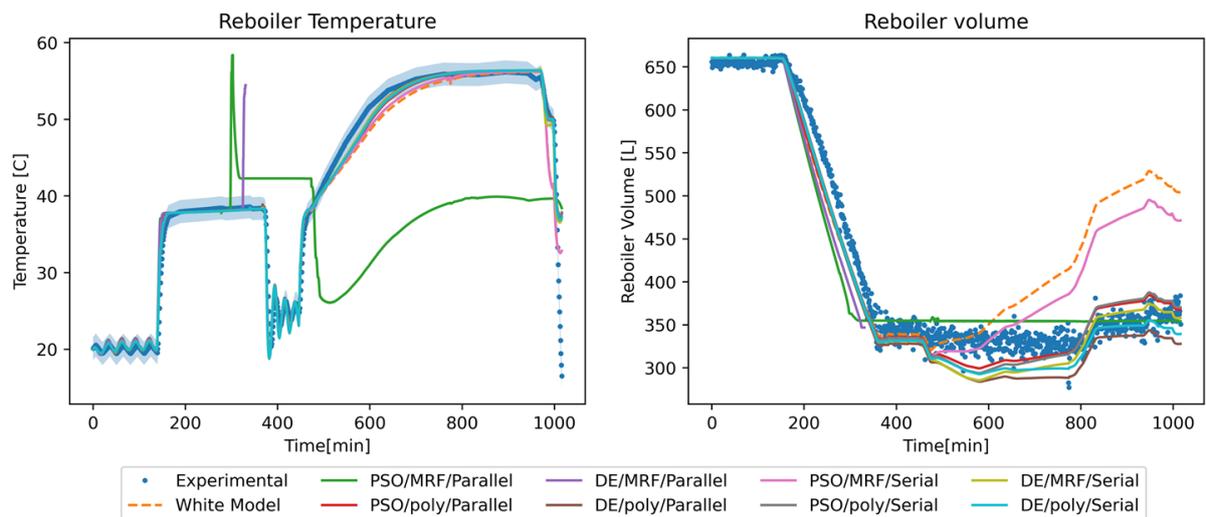


Figure 1s: Overall view of the predicted process variable for the test batch at the random state that return the lower fitness function value for the technique compared with the experimental values. The PSO/MRF/Parallel is the weakest technique since the strong deviation on the reboiler temperature and the meaning less profile in the reboiler volume. The polynomial function shows better performance than the MRF function, especially in predicting the volume inside the reboiler. However, all the techniques perform better than the optimized white model

The PSO/MRF/Parallel and the DE/MRF/Parallel techniques are highly influenced by the structure of the fitness function because the MRF structure increases the non-linearity of the parameter search that make the exploration of new solution more complex since the relation between the parameter changing and the changing in the fitness function is strongly non-linear.

The PSO/Poly/Parallel and the DE/Poly/Parallel show augmented prediction accuracy compared to the solutions obtained using the MRF. This is related to the increased prediction accuracy and robustness in the h_{dev}^{HM} profile returned by the Poly/Parallel techniques compared to the results returned by the MRF/Parallel approaches. In Table 1s all the determination coefficients returned by the PSO/MRF/Parallel have lower values than the ones associated to the polynomial structure.

The usage of the serial training approach does not deliver any observed improvement on the accuracy of the MRF based models; also in this case the predictions from the hybrid model based on polynomial structure outstand the ones based on MRF in following the experimental data. The DE/Poly/Serial solution shows better predictions in the volume profile compared to all the other techniques.

Table 1s: Determination coefficient (R^2) of the profile within the reboiler predicted by the various training technique and approaches for the test batch and the extrapolation batch

Technique	Test batch			
	Reboiler Temperature		Reboiler Volume	
	Parallel	Serial	Parallel	Serial
PSO/MRF	-0.081	0.978	0.940	0.778
PSO/poly	0.981	0.983	0.970	0.969
DE/MRF	0.939	0.984	0.789	0.962
DE/poly	0.984	0.983	0.948	0.966
FP model	0.975		0.631	

Table 1s highlights the increased prediction accuracy of the hybrid models compared to the white-box approach. The increased prediction capabilities affect mostly the amount of liquid within the reboiler and has only a marginal impact on the temperature. This is related to the physic of the system; the temperature is linked to the boiling temperature of the mixture inside the reboiler. The concentration inside the reboiler is influenced by the evaporated solvent blend, however the impact of this parameter higher on the reboiler volume than the temperature.

Beside the increased prediction accuracy on the test set, the hybrid models based on polynomial function also report high extrapolation capabilities. They follow the process variables obtained also from a batch that works with half of the volume of the training batch. This can be observed in the right side of Figure 2s in which the polynomial functions are still able to follow the experimental data. An interesting result is the stabilization of the temperature profile in the PSO/MRF/Parallel and DE/MRF/Parallel that in the batch where the volume is the half of the original one returns a solution that follows the physical trend of the temperature profile inside the reboiler. This could be related to the volume inside the reboiler that affect the k_{UA}^{HM} model and stabilize the solver. Further investigations are needed to explore the reported extrapolation phenomena of the PSO/MRF/Parallel and DE/MRF/Parallel. The extrapolation capabilities are one of the main advantages of the hybrid model, however they are possible only by choosing the proper black-box function technique. From Table 4 is possible to assess the quality of the extrapolation. All the hybrid models have increased prediction capabilities compared to the white-box model. Among the hybrid models the DE/poly/Parallel performs the better.

The authors want to highlight that the differences in the volume prediction capabilities between the various techniques and approaches for the extrapolation batch is more pronounced than what is reported in Table 1s. As shown in Figure 2s, during the operation a premature interruption of the volume measurement occurred. It takes place in the area where the different models differ more and the values after the stop of the volume measurement are not utilized to calculate the determination coefficient.

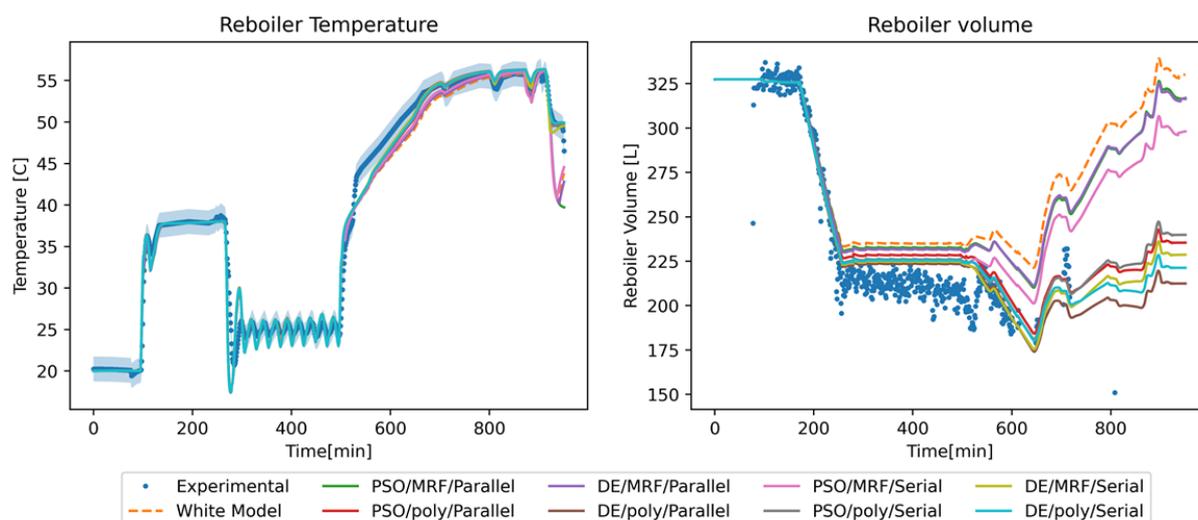


Figure 2s: Overall view of the extrapolation capabilities of the models obtained with the various techniques and approaches. The MRF performs worse than the polynomial in predicting the displayed variables. The usage of different techniques and optimizer does not improve the prediction accuracy of the model based on MRF functions.

s3. Reference

- Amer, H.H., Paxton, R.R., Winkle, M. Van, 1956. Methanol-Ethanol-Acetone. *Ind. Eng. Chem.* 48, 142–146. <https://doi.org/10.1021/ie50553a041>
- Khurma, J.R., Muthu, O., Munjal, S., Smith, B.D., 1983. Total-pressure vapor-liquid equilibrium data for binary systems of dichloromethane with pentane, acetone, ethyl acetate, methanol, and acetonitrile. *J. Chem. Eng. Data* 28, 412–419. <https://doi.org/10.1021/je00034a020>
- Martin, M.C., Cocero, M.J., Mato, F., 1991. Vapor-liquid equilibrium data at 25°C for six binary systems containing methyl acetate or methanol, with dichloromethane, chloroform, or 1,2-trans-dichloroethylene. *J. Solution Chem.* 20, 87–95. <https://doi.org/10.1007/BF00651642>
- Miranda, L.J., 2018. PySwarms: a research toolkit for Particle Swarm Optimization in Python. *J. Open Source Softw.* 3, 433. <https://doi.org/10.21105/joss.00433>
- Nath, J., Prakash Dixit, A., 1990. Total vapour pressures for binary liquid mixtures of acetone with tetrachloroethylene, trichloroethylene, methylene chloride, 1,2-dichloroethane and cyclohexane at 273.15 K. *Fluid Phase Equilib.* 60, 205–212. [https://doi.org/10.1016/0378-3812\(90\)85052-C](https://doi.org/10.1016/0378-3812(90)85052-C)
- Renon, H., Prausnitz, J.M., 1968. Local compositions in thermodynamic excess functions for liquid mixtures. *AIChE J.* 14, 135–144. <https://doi.org/https://doi.org/10.1002/aic.690140124>
- Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, \.Ilhan, Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors, 2020. {SciPy} 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>