# Determining Robust Reaction Kinetics from Limited Data

Gizem Ozbuyukkaya, Robert S. Parker, and Goetz Veser

Department of Chemical Engineering, Swanson School of Engineering, and Center for
Energy, University of Pittsburgh, Pittsburgh, PA 15261

**Abstract:** Accurate chemical kinetics are essential for reactor design and operation. However, despite recent advances in "big data" approaches, availability of kinetic data is often limited in industrial practice. Herein, we present a comparative proof-of-concept study for kinetic parameter estimation from limited data. Cross-validation (CV) is implemented to nonlinear least-squares (LS) fitting and evaluated against Markov chain Monte Carlo (MCMC) and genetic algorithm (GA) routines using synthetic data generated from a simple model reaction. As expected, conventional LS is fastest but least accurate in predicting true kinetics. MCMC and GA are effective for larger data sets but tend to overfit to noise for limited data. Cross-validation least-square (LS-CV) strongly outperforms these methods at much reduced computational cost, especially for significant noise. Our findings suggest that implementation of cross-validation with conventional regression provides an efficient approach to kinetic parameter estimation with high accuracy, robustness against noise, and only minimal increase in complexity.

**Topical heading:** Reaction Engineering, Kinetics and Catalysis

**Keywords:** Reactor modeling, limited data, overfitting, cross-validation

## 1. Introduction

Precise knowledge of kinetic parameters of a chemical reaction system is fundamental not only for improving our understanding of underlying chemical processes, but also for the accurate design, optimization, and robust operation of reactors[1-3]. Kinetic models of reactive systems allow rapid exploration of the reaction outcome over a wide range of operating conditions[4], the design of new experiments, and the synthesis of safety training systems[5,6].

In practice, a kinetic model is typically derived from the fundamental material, energy and momentum balances, including the equilibrium or reaction rates[7-9]. Training of the kinetic model parameters then requires a set of laboratory experiments or availability of data from an operating plant, where the acquisition of data is often expensive or challenging[10-12]. While much focus is currently on high-throughput screening and "big data" approaches, in industrial practice, availability of data is typically limited and often a relatively small number of data points have to suffice to identify kinetic parameters of a chemical reaction[13,14]. Regular nonlinear least-squares fitting is most commonly used in kinetic fitting of the experimental data[15-17]. However, the method is well-known to be sensitive to noise and, due to the deterministic nature of the method, least-squares estimation can get stuck at local minima[18,19]. To circumvent this issue, more sophisticated optimization methods such as Markov chain Monte Carlo (MCMC), or genetic algorithm (GA) are used to explore parameter spaces[4,20,21]. MCMC performs a random walk in parameter space and may accept "bad" moves probabilistically (movements in the direction of increasing objective function) to escape local minima[22,23]. GA, on the other hand, is a population-based algorithm that performs parameter estimation based on the "survival of the fittest" in real life evolution[24,25]. However, both of these methods require expert knowledge and higher computation cost, although all implementations of GA and some implementations of MCMC are parallelizable[26].

In recent decades, significant advancements have been made towards building supervised learning models from big data for fault detection, process modeling, and control of chemical reactions[27-30]. In contrast, in the case of limited data availability, which is still typical in industrial practice, statistical validation of model-based reaction kinetics is often overlooked. Regardless of the choice of algorithm, the accuracy of the kinetic prediction is often assessed by the error value of the regression, which shows how well the model parameters match the experimental data[31,32]. This approach, however, may result in an overfit, in which the model "memorizes" the specific set of data, rather than "learning" the underlying trends[29,33,34]. This severely limits the capability of the model to truly predict system behavior. Moreover, complex models are more likely to overfit when data is limited[32]. To prevent overfitting, models need to be validated on "fresh" data they have not seen yet[35,36]. For this purpose, a portion of the experimental data, the "hold-out" (ideally 10-30% of the available data), is set aside to be used for validating after the model is trained[37,38]. However, it can be challenging to hold out or exclude a fraction of the available data from model training, particularly if availability of data is already limited. Besides, the selection of data for hold-out may not be straightforward. For example, if the noise level or the number of outliers present in the validation data is significantly higher than in the training data, model assessment can become highly inaccurate[36].

As a solution to this problem, a cross-validation (CV) methodology can be applied to the kinetic parameter regression[39]. This method is commonly used in classification problems in machine learning for accuracy reporting, in which the prediction output belongs to a discrete set of categories or classes[40-42]. CV, similar to the "hold-out" method described above, is based on the splitting of the experimental data into two sets: a "training set" to build and predict the model parameters, and a "validation set" to assess the quality of the model. For this purpose, the experimental dataset is partitioned into k nearly equal-sized subsets or "folds". The model is trained using (k-1) folds, and the accuracy of the model is then validated on the fold that was left out (i.e., the k$^{th}$ fold). This step is repeated k times for each possible permutation of folds with a different subset left out each time as the validation fold. Upon completion, kinetic parameters from all runs are then averaged based on the error of their respective validation subset[39].

In contrast to classification problems and machine learning, CV has not been broadly applied nonlinear regression problems or kinetic parameter estimation to-date[30,43-45]. Yet, it can be implemented as a straightforward extension of the conventional nonlinear least-squares fitting procedure and can be expected to yield significant benefit towards obtaining robust kinetics. This is particularly true for cases with limited data availability, since CV ensures that the model is ultimately trained with each data point, and, similarly, each data point has a chance of being validated against the model parameters. In addition, due to the statistical averaging of the successive runs, the possibility of an overfit is minimized, improving the algorithm's true predictive ability.

In the present work, a proof-of-concept study compares different algorithms for robust kinetic parameter estimation in the presence of limited data. Specifically, a combination cross-validation plus nonlinear least-squares fitting routine is compared to stand-alone nonlinear least-squares[46], MCMC[23], and GA[24] algorithms for kinetic parameter estimation. The analyses are performed on synthetic data for the purpose of probing dataset properties such as size (number of experimental points), noise level, and number of outliers. The water-gas shift reaction with simple, well-established lumped kinetics is used as the basis for a simple one-step model reaction[47]. Synthetic data is generated by simulating the kinetic model over a series of operating conditions and adding controlled levels of Gaussian white noise and outliers to the generated data points. Method performance is critically assessed in terms of both accuracy (prediction accuracy of the true reaction kinetics) and numerical efficiency (number of function evaluations). The overall aim of the study is to yield guidelines for the practitioner towards improved, robust kinetic data fitting without requiring advanced training in mathematical methods.

## 2. Computational Methods

### 2.1 Model Construction

A simple isothermal steady-state kinetic plug-flow reactor model is constructed using conservation of mass equations and typical Arrhenius-type reaction kinetics:

$$\frac{\partial Cj}{\partial t}=0=\frac{-1}{A_{CS}}\cdot\frac{\partial \dot{n}_j}{\partial z}+\frac{m_{Cat}}{V_R}\cdot\sum_i v_{ij}\cdot r_i \tag{1}$$

$$\text{at } z=0\text{: } c_j=c_{j,inlet},\quad \text{at } z=\text{L: } \frac{\partial c_j}{\partial z}=0 \tag{2}$$

Here $C_j$ is the concentration of each component j (mol/m$^3$), $A_{CS}$ is the cross-sectional area of the tubular reactor (m$^2$), $\dot{n}$ is the molar flowrate of each component (mol/s), $m_{cat}$ is the catalyst weight (kg), $V_R$ is the volume of the reactor (m$^3$), $r_i$ is the rate of each reaction i (mol/kg$_{cat}$/s) and $v$ is the stoichiometric coefficient of each component. Conversion values are calculated with the following equation:

$$Conversion\text{, }species\ i\ X_i=\frac{C_{i,0}-C_i}{C_{i,0}} \tag{3}$$

In this work, water-gas shift is chosen as a simple, industrially relevant model reaction with well-known kinetics in the literature[47]:

WGS: $CO + H_2O \rightleftharpoons CO_2 + H_2$

$$r_{WGS} = k_{0,f} \cdot \exp\left(\frac{-E_{A,f}}{R \cdot T}\right) \cdot [CO][H_2O] - k_{0,b} \cdot \exp\left(\frac{-E_{A,b}}{R \cdot T}\right) \cdot [CO_2][H_2] \qquad (4)$$

Where $r_{WGS}$ denotes the (net) rate of reaction (mol/kg$_{cat}$/s), $k_{0,f}$ and $k_{0,b}$ denote the pre-exponential factors, and $E_{A,f}$ and $E_{A,b}$ the activation energy for the forward and back reaction, respectively, and [CO], [$CO_2$], [$H_2$] and [$H_2O$] denote the concentrations of the respective reactants and products.

However, an initial model evaluation revealed that one of the model parameters, $k_{0,b}$, is not identifiable (explained further below in "profile likelihood" subsection). Hence, the value of $k_{0,b}$ was increased by an order of magnitude in this study to ensure that all parameters are identifiable. Therefore, the model reaction is presented as a generic reaction throughout the paper $(A + B \rightleftharpoons C + D)$, with parameters given in Table 1. Synthetic data is generated by using kinetic parameters and reactor geometry for the model reaction and solving the material balance equations to determine the concentration of each species at the reactor exit.

**Table 1 Kinetic parameters and reactor specifications of the model reaction**

| **Parameters** | | | **Specifications** | | |
|---|---|---|---|---|---|
| **$k_{0,f}$** | $3.0 \times 10^5$ | molgcat$^{-1}$ h$^{-1}$atm$^{-2}$ | **tube dia.** | 0.5 | inch |
| **$k_{0,b}$** | $2.5 \times 10^8$ | molgcat$^{-1}$ h$^{-1}$atm$^{-2}$ | **tube length** | 1 | ft |
| **$E_{a,f}$** | $5.0 \times 10^4$ | J/mol | **catalyst wt.** | 1 | kg |

Varying sizes of experimental datasets (inlet and outlet concentration values) are generated over the following parameter ranges: temperature (150°C, 175°C, 200°C, 225°C and 250°C), molar inlet A to B ratio (0.25, 0.5, 1, 2 and 4), and gas hourly space velocity (GHSV) ($3.0\text{x}10^3 h^{-1}$, $6.0\text{x}10^3 h^{-1}$, $1.2\text{x}10^4 h^{-1}$, $1.8\text{x}10^4\ h^{-1}$, $2.4\text{x}10^4\ h^{-1}$). After synthetic data is generated, varying levels of Gaussian white noise are added to concentration values. If the added noise generates a concentration that is out of physical range (e.g., negative values), the noise is recalculated until it satisfies the physical constraints.

- *Local sensitivity*

Following model construction and data generation, sensitivity analysis is carried out to determine how strongly a given parameter and the model outcome are correlated[48,49]. The analysis is performed by calculating finite-difference based sensitivity coefficients, the result of applying a small change to one parameter at a time for a given axial displacement (z):

$$s_{i,j}(z)=\frac{\partial x_i}{\partial \theta_j}=\frac{x_i(\theta_j+\Delta\theta_j,z)-x_i(\theta_j,z)}{\Delta\theta_j} \qquad (5)$$

Here θ is the fit parameter, and x is the dependent variable (e.g., concentration). These sensitivity coefficients are then normalized for direct comparison:

$$\overline{s_{ij}}(z)=s_{i,j}(z)\times\frac{\theta_j}{x_i} \qquad (6)$$

Finally, relative sensitivity (RS) is calculated for each parameter:

$$RS_{i,j}=\frac{1}{Q_Z}\sqrt{\sum_{k=1}^{Q_z}\left|\overline{s_{i,j}}(z_k)\right|^2} \qquad (7)$$

Here $z_k$ ($k \in [1, Q_Z]$) are discrete axial displacements where concentrations are calculated and $Q_Z$ is the total number of calculations.

- *Profile likelihood*

Finally, profile likelihood analysis is performed to determine if model parameters are identifiable with the generated experimental data set. For this purpose, a log-likelihood (LL) value is calculated from model fitting, which is proportional to the normalized sum of square errors[50]:

$$\text{¿} \cong -\frac{1}{2}\sum_{i}\frac{\left(y_i - g(z,\theta)\right)^2}{\sigma^2} \tag{8}$$

Here $y_i$ is the experimental data (e.g., concentration), g is the model-predicted output at independent variable z (axial displacement) calculated with fit parameter set $\theta$, and $\sigma$ is the standard deviation of the experimental data. Profile likelihood is the maximum log-likelihood, i.e., the value of LL when the following objective function is minimized:

$$PL_j(p) = \max_{\theta \in (\theta \vee \theta_j = p)} \text{¿}(y \vee \theta) \tag{9}$$

Here p is the fixed parameter value, LL is the log-likelihood and $\theta$ is the parameter set. The profile-likelihood (PL) is calculated by optimizing fit parameters while keeping one parameter at a time, $\theta_j$, at a pre-set value p. The optimization is repeated for a discrete range of the fixed parameter ($\theta_j \in [\theta_{j,min}, \theta_{j,max}]$) and PL values of each estimation are plotted. If the resulting PL landscape does have a unique minimum, and exceeds the 95% confidence threshold, the parameter is termed identifiable and estimable from the experimental data[50].

### 2.2 Parameter estimation

In order to assess the predictive ability of each parameter estimation method, 20% of the synthetic data is held out for external testing, and the remaining 80% of the data is used towards model training. For the purpose of distributing the synthetic data evenly between training and test sets, stratified sampling is carried out based on input conditions (e.g., temperature) instead of random sampling. After adding the same level of noise to each set, the model is trained with the generated data (Figure 1).

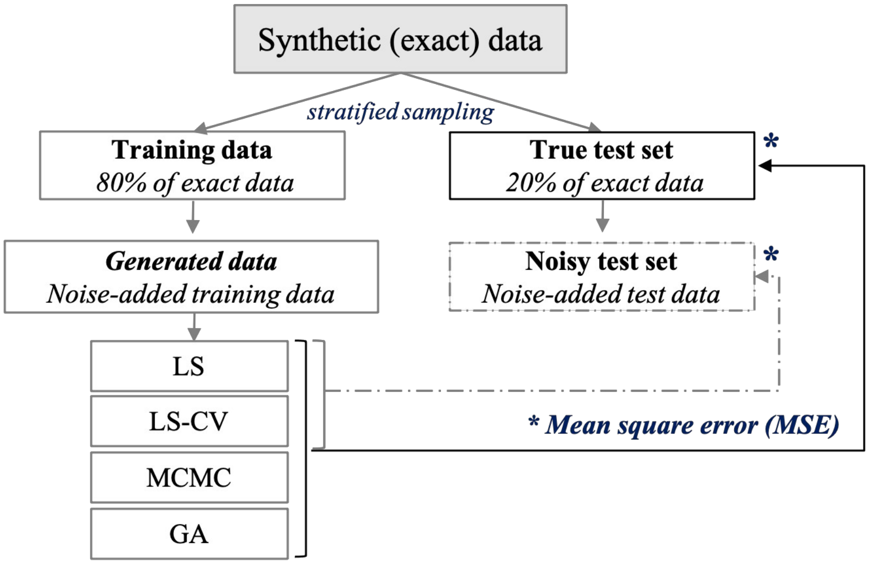

**Figure 1 The simplified schematic of the computational workflow**

Agreement with true kinetics is determined by calculating the mean square error (MSE) of the "true test set", which is generated from the actual kinetic parameters. Comparison of estimated and real parameters is also tabulated in Table S1-S3. In addition, the MSE of the "noisy test set" is reported as a representative metric of a real-life scenario where the true kinetics are not known, and only noisy experimental data will be available for testing. The number of function evaluations is also used to directly compare the computational cost. For statistical comparison of all methods, for each dataset size and noise level combination, 50 repeat parameter estimations are carried out on the same generated data. T-tests are performed on the MSE values of the test sets and the number of function evaluations from each method. For more in-depth comparison of LS and LS-CV methods, where a different dataset is generated for each repeat run, paired t-tests are performed. Prior to the analyses, the normality assumption of t-test was checked by Kolmogorov-Smirnov test, demonstrating that t-test was the appropriate statistical analysis[51,52].

*Nonlinear least-squares (LS):* The LS regression is performed using MATLAB's *lsqnonlin* function based on Levenberg-Marquardt (LM) method with default tolerance values[46] and the following objective function:

$$Objective\ function = \sum_{i=1}^{m} \sum_{j=1}^{n} \left( C_{i,j}^{\exp} - C_{i,j}^{Model} \right)^2 \tag{10}$$

Here C refers to the concentrations, i refers to the components and j refers to the experimental (or generated) data point for each component.

*Cross-validation nonlinear least-squares (LS-CV):* For LS-CV, data is divided into folds via stratified sampling using the *crosvalind* function in MATLAB. Stratified sampling assures that the distribution of the input conditions (e.g., temperature) on the overall dataset is represented similarly on the individual folds[53]. Within each run, a different fold of the data is left out for validation while the remaining folds are used for training the model using nonlinear least-squares fitting[54,55]. The runs are repeated k times (the number of folds). As a result, an ensemble of parameters and the associated validation errors are obtained from each run, which are then weighted based on their respective validation errors to determine overall fit parameters:

$$\Theta = \sum_{i=1}^{k} \omega_i \theta_i, \quad \omega_i = \frac{1/E_{i,test}^2}{\sum_{i=1}^{k} 1/E_{i,test}^2} \tag{11}$$

Here θ is the fitted parameters of each run, ω is validation error weight, E is the validation error of each run, and k is the number of folds, which was chosen to be 5 for a dataset size of 25, and 10 for larger dataset sizes.

*Markov chain Monte Carlo (MCMC):* MCMC is performed using the Metropolis Hasting algorithm[56]. The random walks are sampled from a log-normal distribution for pre-exponential factors and from a normal distribution with a standard deviation of 5000 kJ/mol for activation energies. Each move that reduces the objective function is accepted. Moves that increase the objective function are accepted if the calculated Metropolis criterion is larger than a uniformly generated random number. The Metropolis criterion is calculated using the following formula:

$$\beta = e^{\frac{-\left(SSE_{new} - SSE_{old}\right)}{2\sigma^2 T}} \tag{12}$$

Here $\beta$ is the Metropolis criterion, SSE is the sum of squared error, $\sigma^2$ is the variance of the SSE values and T is the temperature factor, which adjusts the stringency of the algorithm and is selected as 3 for dataset size of 25 and 5 for larger dataset sizes. Metropolis Hasting is implemented with a simulated annealing algorithm, which slowly decreases the probability of accepting worse solutions as the solution space is explored[57,58]. Based on 20 initial realizations performed using three different noise level and data set size combinations with 50000 iterations, the maximum number of iterations is selected as 20000 since no improvement after the 20000th step was observed. From the analysis on the same set of realizations, the runs are terminated if the best objective function does not change for 2000 steps.

*Genetic algorithm (GA):* The GA generates a group of candidate parameters (population), which are generated from a given parameter space with Latin hypercube sampling[59]. The initial population size is selected as 400, and the elitism parameter is selected as 0.1, which discards the parameter candidates that rank below the top 10% based on the objective function. The elite population is used to generate a new population using a “crossover” procedure, in which the parameters of two randomly selected samples from the elite population are averaged with random weights generated from standard uniform distribution to create new offspring. This procedure is repeated until population size reaches 400. Lastly, 10% of candidate parameters are subjected to mutation, which introduces diversity into the population by allowing random walk (factored by a log-normal random variable) of the parameter space. The optimization is terminated if the best objective function does not improve for 20 generations.

## 3. Result and Discussion

### 3.1 Model Evaluation

Initially, synthetic data are generated by simulating the constructed reaction model at steady state conditions with varying operating conditions, and by adding controlled levels of noise, as described in Section 2.1. Ranges of these operating conditions are selected to ensure that the data covers a wide range of conversion values to avoid a highly localized kinetic model. Representative noise-added generated data, along with the distribution of conversion, are shown in Figure 2. Prior to parameter estimation, local sensitivity and profile likelihood analysis are performed on the kinetic model to ensure that the model parameters are identifiable and have a significant effect on the model outcome, i.e., that the estimation problem is well posed. In practice, kinetic modeling studies are typically conducted under the assumption that the model parameters are identifiable based on the available experimental data without verifying this assumption. These pre-analysis tools could be useful for experimentalists and practitioners to establish confidence in fitted parameters, model reduction, and the design of experiments[60]. Accordingly, the calculated relative sensitivities, which are expressed as the percent change in output concentrations of all species due to a slight change (1%) in model parameters, are shown in Table 2. Model outputs are sensitive to all kinetic parameters, activation energy having the highest relative sensitivity due to the exponential relation in the Arrhenius equation.

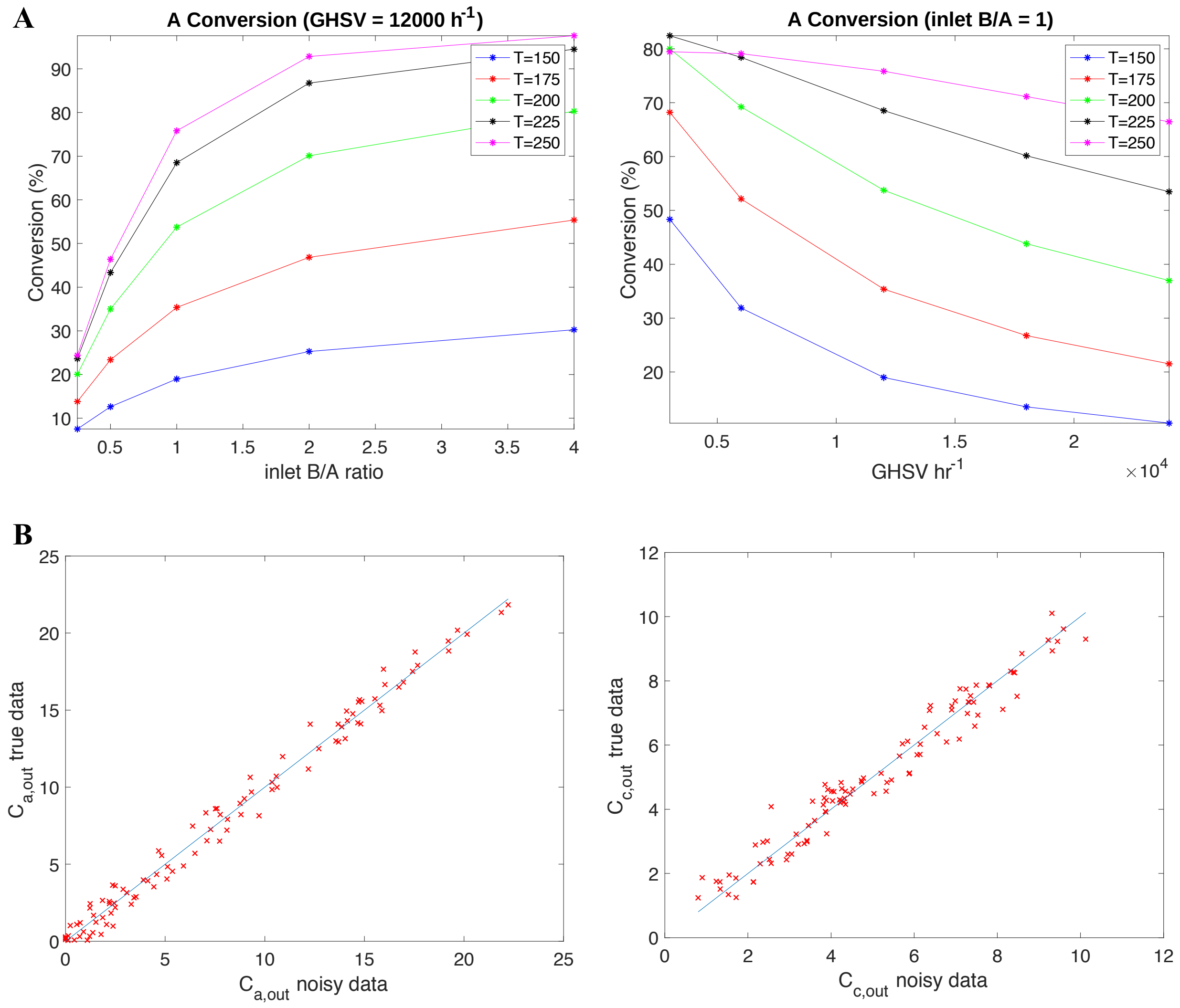

**Figure 2 A) The distribution of species A conversion under varying operating conditions (inlet temperature, input ratios and GHSV values), and B) Representative generated concentration data of species A and species C (dataset size: 100, noise level:0.1)**

**Table 2 Relative sensitivities of kinetic parameters to output variables**

| **Relative Sensitivity (%)** | **$C_A$** | **$C_B$** | **$C_C$** | **$C_D$** |
|---|---|---|---|---|
| **$k_{0,fwd}$** | 7.47 | 7.47 | 7.43 | 7.43 |
| **$k_{0,bwd}$** | 6.27 | 6.27 | 3.50 | 3.50 |
| **$E_{A,fwd}$** | 86.26 | 86.26 | 89.07 | 89.07 |

Similarly, likelihood profiles are generated using representative generated data, which are given in Figure 3. The presence of clearly defined minima in the identifiability plots for the pre-exponential factors and the activation energy confirms that the model parameters are estimable within the temperature range tested here (150-250°C).

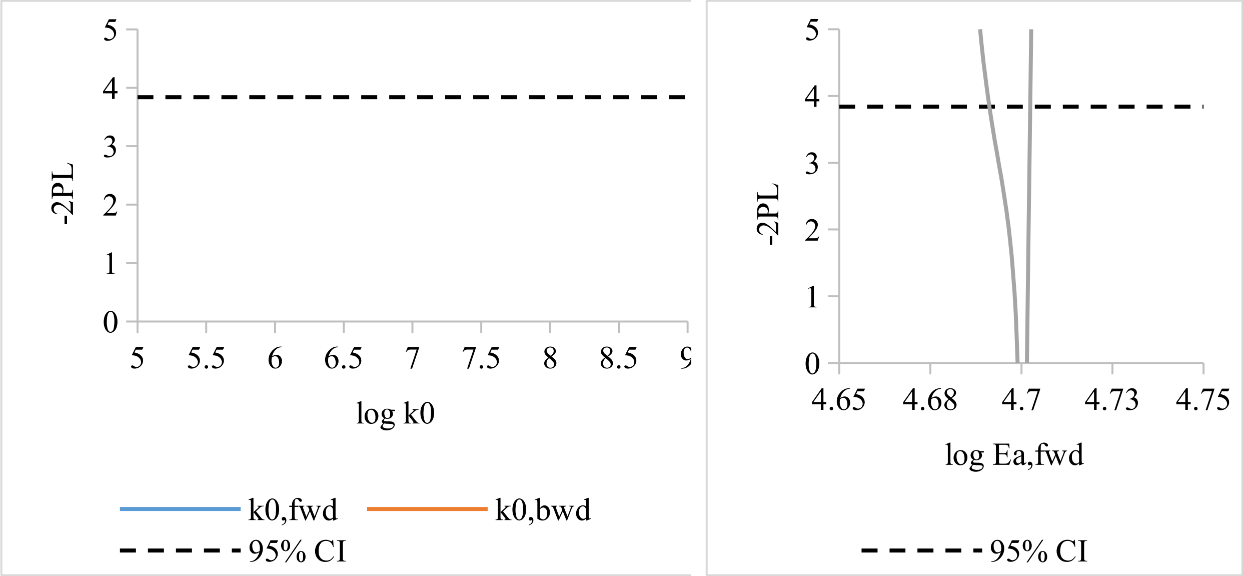

**Figure 3 Profile likelihood plots for the model parameters (For plotting purposes, the minimum of -2PL is subtracted from all -2PL values.)**

### 3.2 Accuracy and Efficiency Comparison of All Methods

In the first stage of our comparative study, all four methods (LS, LS-CV, MCMC, and GA) are comparatively evaluated with regard to their accuracy and computational efficiency. This comparative parameter estimation study is carried out using three different pairs of dataset size (n) and noise levels (nl) to observe the effect of dataset properties on algorithm performances. All 3 operating variables can take on 5 discrete values (Figure 2A), resulting in a maximum dataset size of 125 datapoints. Three subset sizes (n = 25, 50, and 100) were chosen, each with a different noise level (decreasing from nl=0.5 for the smallest dataset to nl = 0.1 for the largest one), resulting in the following combinations for the analysis: data set 1: n=25, nl=0.5; dataset 2: n=50, nl=0.2; dataset 3: n= 100, nl=0.1. The accuracy of the methods is determined based on their closeness to the true kinetics, as indicated by the MSE against the test set (Figure 4). In addition, the efficiency or computational cost is determined by the number of function evaluations required to derive the parameters (Figure 5). For stochastic methods (MCMC, GA, LS-CV), 50 repeat runs (using different folding splits) are carried out on the same generated dataset and training-test split for direct comparison and the statistical significance is reported by p-values (Table 3 and Table 4).

Kinetic accuracy comparison of all methods with different dataset size and noise level combinations are shown in Figure 4. Distribution of test set MSE values for LS-CV, MCMC and GA methods are shown in boxplots, in which mean and median values are shown by "x" and "–", respectively. The interquartile range is shown by box, whereas the outliers are denoted by the dots. The list of derived kinetic parameters is given in Table S1-S3. As apparent in the MSE distribution plots, LS-CV and the stochastic methods are able to predict the true kinetics with comparable accuracy for the cases with a larger number of datapoints (n:50,100) and lower noise levels (nl:0.1, 0.2), and all three outperform the simple nonlinear least-squares (LS) method (Figure 4A-B). LS-CV performs similarly to the random search methods due to its use of multiple local solutions (one per fold) that are statistically averaged to provide a composite parameter estimate. Slight differences in mean errors are observed for LS-CV, MCMC and GA methods, but they are not statistically significant for these datasets (Table 3). In contrast, for the runs with a smaller number of datapoints and relatively higher noise (n:25, nl:0.5), LS-CV significantly outperforms both MCMC and GA methods in terms of accuracy of true kinetics (see Figure 4C). In these runs, both MCMC and GA had the lower training errors amongst all methods, i.e. they appear to fit to the data much better. However, since these methods also overfitted to experimental noise in the data, they failed to capture underlying kinetics, hence higher MSE on the true parameter test set. On the other hand, LS-CV implementation efficiently filters out noise in the data due to the model validation, which is based on all possible permutations of the (noisy) data.

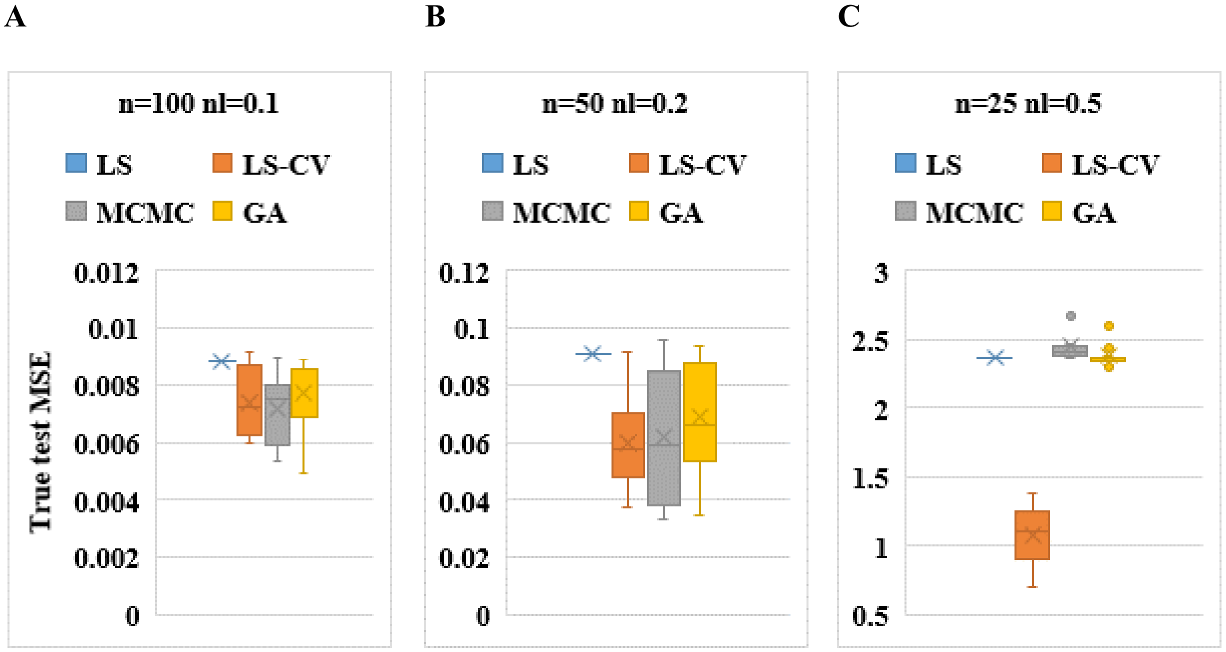

**Figure 4 Kinetic accuracy (test set MSE) comparison of all methods with different dataset size and noise level combinations (A: n=25, nl=0.5; B: n=50, nl=0.2; C: n= 100, nl=0.1). Mean and median values are denoted by "x" and "–", respectively. Note that the magnitudes of y-axes are different.**

The number of function evaluations necessary to perform the parameter estimation task is used to compare the computational efficiency of the four different methods. As expected, LS requires the lowest number of function evaluations due to the deterministic nature of the optimization algorithm. The total number of objective function evaluations is increased significantly for more complex algorithms, MCMC and GA, due to their stochastic nature, and directed random walk strategies. Due to population size, GA requires the highest number of function evaluations, although estimations with independent candidate parameter sets could be run in parallel. LS-CV, although less efficient than regular LS, requires 3- to 10-fold fewer function evaluations compared to MCMC and GA. Similarly, an order of magnitude increase in run time from LS-CV to the probabilistic methods is also observed (all runs were performed on Intel(R) Core(TM) i7 3.70GHz, 6 Cores.) This increased efficiency of LS-CV is found consistently across the datasets, as confirmed by a t-test performed on different data set size and noise level combinations (Table 4). This suggests that CV in combination with LS can provide a straightforward solution to parameter estimation problems by adding minimal complexity while obtaining comparable or better accuracy than more complex stochastic methods.

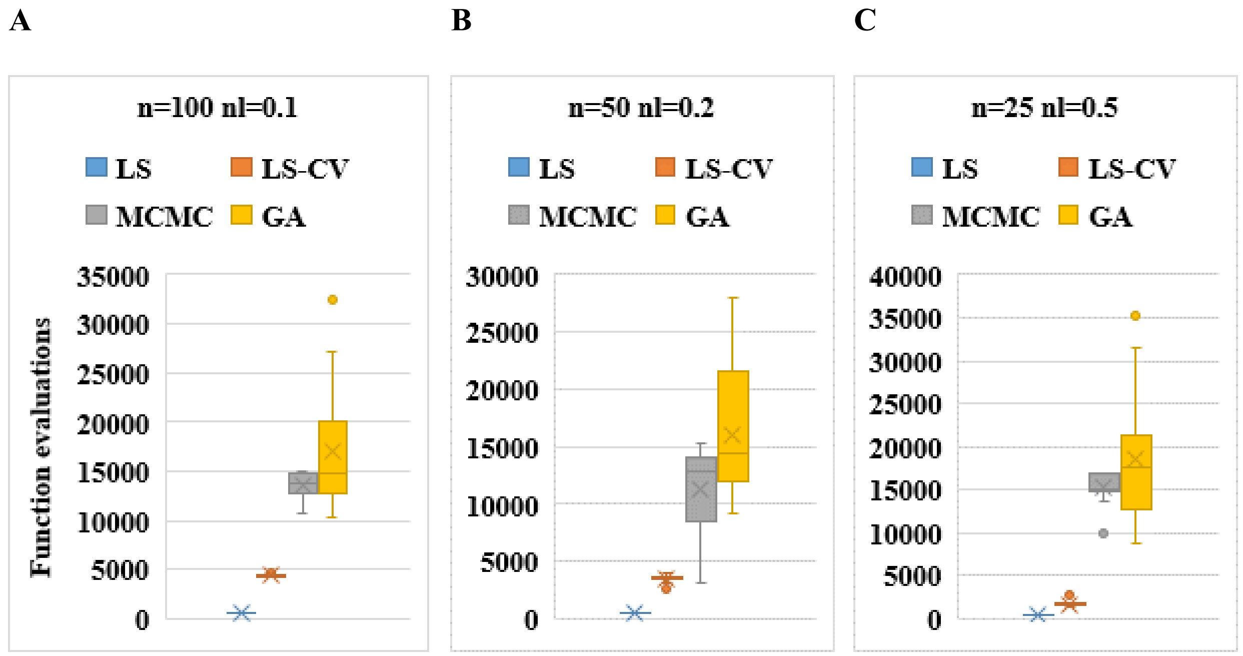

**Figure 5 Computational cost (number of function evaluations) comparison of all methods with different dataset size and noise level combinations (A: n=25, nl=0.5; B: n=50, nl=0.2; C: n= 100, nl=0.1). Mean and median values are denoted by "x" and "–", respectively.**

**Table 3 Statistical analysis of kinetic accuracy (test set MSE) comparison of all methods with different dataset size and noise level combinations (p-value code: *<<0.00001)**

| | **n=100 nl= 0.1** | | | **n=50 nl= 0.2** | | | **n=25 nl= 0.5** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **LS-CV** | **MCMC** | **GA** | **LS-CV** | **MCMC** | **GA** | **LS-CV** | **MCMC** | **GA** |
| **mean** | 0.0074 | 0.0071 | 0.0077 | 0.0596 | 0.0622 | 0.0690 | 1.0789 | 2.3916 | 2.2023 |
| **std dev** | $1.23\text{x}10^{-3}$ | $1.18\text{x}10^{-3}$ | $1.30\text{x}10^{-3}$ | $1.42\text{x}10^{-2}$ | $2.46\text{x}10^{-2}$ | $1.94\text{x}10^{-2}$ | $2.10\text{x}10^{-1}$ | $9.20\text{x}10^{-2}$ | $2.58\text{x}10^{-1}$ |
| **p-value** | | >0.05 | >0.05 | | >0.05 | >0.05 | | * | * |

**Table 4 Statistical analysis of computational cost (number of function evaluations) comparison of all methods with different dataset size and noise level combinations (p-value code: *<<0.00001)**

| | **n=100 nl= 0.1** | | | **n=50 nl= 0.2** | | | **n=25 nl= 0.5** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **LS-CV** | **MCMC** | **GA** | **LS-CV** | **MCMC** | **GA** | **LS-CV** | **MCMC** | **GA** |
| **mean** | 4381 | 13657 | 17089 | 3494 | 11212 | 15973 | 1652 | 15198 | 18582 |
| **std dev** | 132 | 1314 | 5975 | 259 | 3972 | 5452 | 285 | 1648 | 6960 |
| **p-value** | | * | * | | * | * | | * | * |

### 3.3 Effect of Dataset Size, Noise and Outliers

Based on the above comparison between the three advanced methods (LS-CV, MCMC, and GA), LS-CV is selected for a more detailed investigation of the effect of dataset properties, namely dataset size, noise, and outliers, in comparison to simple LS. Performance comparisons are conducted on independently generated datasets for each run to ensure that results are reproducible on different datasets. In other words, using the same kinetic model and noise level, multiple random dataset generations and training-test splits are performed. Hence, paired t-tests are carried out to evaluate statistical significance after conducting Kolmogorov-Smirnov Test of

Normality[51,52] to confirm data distribution can be assumed to be normal. The analyses are conducted with all possible combinations of three levels of dataset sizes (25,50,100) and four levels of noise (0.1, 0.2, 0.3, 0.5). The accuracy of the parameter estimation, similar to the previous study, is determined based on the MSE against the test set.

Kinetic accuracy (test set MSE) comparison of LS and LS-CV methods with increasing level of noise and dataset size is illustrated in Figure 6 by a representative run whereas the statistical analysis of all repeat runs is shown in Table 5. The list of derived kinetic parameters is given in Table S1-S3Error: Reference source not found. Accordingly, LS-CV outperforms the regular LS fitting in all cases with varying levels of improvement. Paired t-tests for all runs show statistical significance of the results ($p < 0.01$).

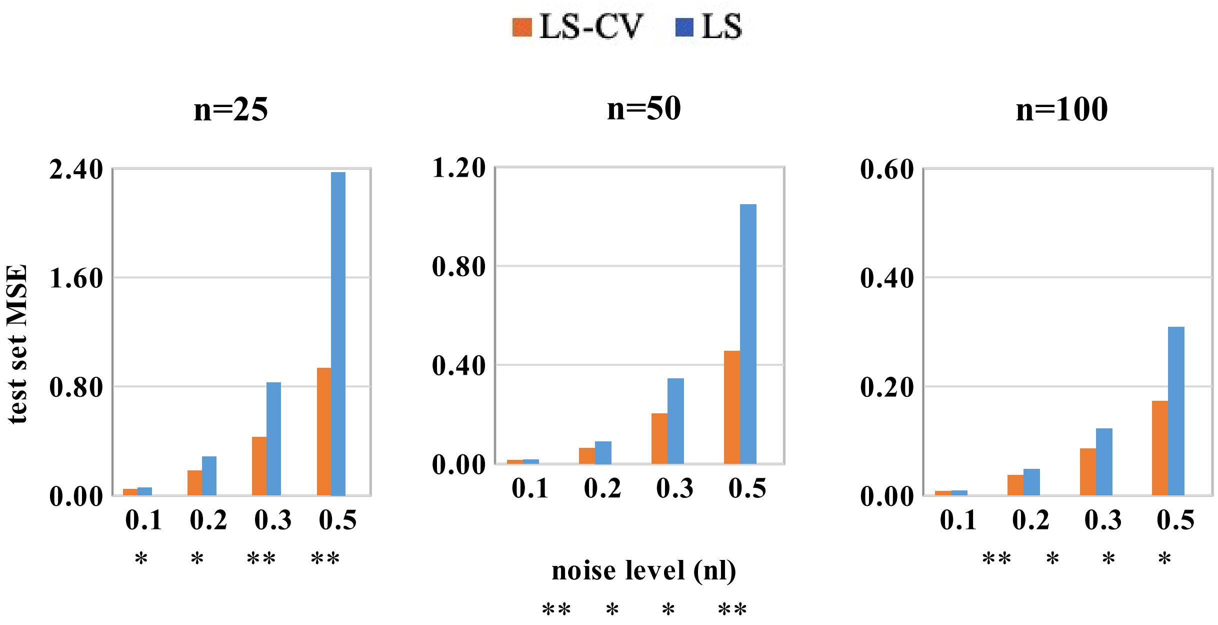

**Figure 6 Kinetic accuracy (test set MSE) comparison of LS and LS-CV methods with all dataset size and noise level combinations. Note that different magnitudes for y-axes are used on the figures. (p-value codes: *<0.0001, **<0.01)**

**Table 5 Statistical analysis of kinetic accuracy (test set MSE) comparison of LS and LS-CV methods with all dataset size and noise level combinations (p-value codes: *<0.0001, **<0.01)**

| | nl | LS | LS-CV | %imp. | p-value |
|---|---|---|---|---|---|
| **n=25** | **0.1** | 0.06 | 0.04 | 20.3% | * |
| | **0.2** | 0.28 | 0.18 | 36.1% | * |
| | **0.3** | 0.83 | 0.43 | 48.3% | ** |
| | **0.5** | 2.37 | 0.94 | 60.6% | ** |
| **n=50** | **0.1** | 0.017 | 0.016 | 9.32% | ** |
| | **0.2** | 0.09 | 0.06 | 30.31% | * |
| | **0.3** | 0.35 | 0.20 | 41.08% | * |
| | **0.5** | 1.05 | 0.46 | 56.43% | ** |
| **n=100** | **0.1** | 0.009 | 0.008 | 7.9% | ** |
| | **0.2** | 0.04 | 0.03 | 23.3% | * |

| | | | | | |
|---|---|---|---|---|---|
| | **0.3** | 0.12 | 0.09 | 30.0% | * |
| | **0.5** | 0.31 | 0.17 | 43.9% | * |

For both methods, the MSE decreases as expected with increasing number of datapoints since more information is available to train the kinetic model. Although LS-CV is consistently more accurate than LS in predicting true kinetics, the relative improvement is more pronounced on limited and more noisy data since LS is prone to overfitting. In LS-CV, this tendency is countered by the fact that the training folds with higher noise will have a relatively larger validation error and hence will be weighted low in the final kinetic parameter value. LS-CV is therefore exceptionally robust against experimental noise. Further comparison of the estimated parameters from these methods shows that LS-CV consistently matches the absolute values of real parameters better compared to LS.

In practice, of course, the "true kinetics" are not available and the goodness of the fit hence needs to be tested against the experimental test data, i.e., the hold-out from the original data set. This data will have the same noise level as the training data (since it is part of the same experimental data set). To mimic this, MSE values of the same runs are also calculated against the noisy test set as a real-life representative metric, where synthetic (non-noisy) data do not exist. Consistent with the findings discussed above, the relative improvement of LS-CV decreases with increasing dataset size for the same noise level, and larger datasets are more tolerant to noise. However, while LS-CV significantly improves the prediction accuracy of true kinetics, it predicts noisy test data only marginally better than simple LS (Table 6). Remarkably, a “naïve observer” would hence rate both methods (almost) equally competent in deriving accurate kinetic parameters, while in fact LS-CV was able to unveil the true kinetics underlying the noisy data with significantly improved accuracy as shown above.

**Table 6 Kinetic accuracy (noisy test MSE) comparison of LS and LS-CV methods with all dataset size and noise level combinations (p-value codes: *<0.005, **<0.05 )**

| **Noisy test MSE** | **n=25** | | | **n=50** | | | **n=100** | | |
|---|---|---|---|---|---|---|---|---|---|
| **nl** | **LS** | **LS-CV** | **p-val.** | **LS** | **LS-CV** | **p-val.** | **LS** | **LS-CV** | **p-val.** |
| **0.1** | 1.75 | 1.74 | ** | 1.73 | 1.73 | * | 1.90 | 1.90 | ** |
| **0.2** | 7.16 | 7.07 | ** | 7.93 | 7.87 | * | 7.88 | 7.86 | * |
| **0.3** | 15.90 | 15.49 | ** | 14.19 | 13.97 | * | 13.17 | 13.11 | ** |
| **0.5** | 44.77 | 41.94 | * | 38.96 | 37.48 | ** | 28.31 | 27.81 | ** |

Experimental data, in particular data from industrial operations, is often subject to outliers, i.e., “bad” data with errors above the general experimental noise level that can be caused by operator error, temporary malfunctioning of sensor, or similar unpredictable effects. To evaluate the ability of LS-CV and LS to handle such outliers, their effect is investigated by progressively increasing the number of outliers in the data. In order to reliably assess method performance against outliers, they are only added to the training data. Outliers are generated by either adding or subtracting one standard deviation of the variable value to the noise-added training data. For all outlier analyses, the same dataset (with same noise points) is used for direct comparison (n:50, nl:0.2) and only outliers change. The MSE against the “true kinetics” test set are tabulated in Table 7. Clearly and unsurprisingly, LS fitting becomes significantly worse at predicting true kinetics with increasing number of outliers. The accuracy of the LS-CV prediction also decreases with more outliers, although the adverse effect is significantly lower compared to LS. Similar to its handling of noise, LS-CV can effectively filter out outliers via statistical averaging.

**Table 7 Effect of outliers in training data on the kinetic accuracy (test set MSE) comparison of LS and LS-CV methods (p-value codes: *<0.0001, **<0.01)**

| | n=50 nl= 0.2<br>no outliers - training | | | | n=50, noise nl = 0.2<br>2 outliers - training | | | |
|---|---|---|---|---|---|---|---|---|
| | **LS** | **LS-CV** | **%imp.** | **p-val.** | **LS** | **LS-CV** | **%imp.** | **p-val.** |
| **true MSE** | 0.091 | 0.064 | 30% | * | 0.115 | 0.066 | 42% | * |
| | **n=50 noise nl = 0.2<br>5 outliers - training** | | | | **n=50, noise nl = 0.2<br>10 outliers - training** | | | |
| | **LS** | **LS-CV** | **%imp.** | **p-val.** | **LS** | **LS-CV** | **%imp.** | **p-val.** |
| **true MSE** | 0.290 | 0.092 | 67% | * | 0.953 | 0.265 | 72% | ** |

**Table 8 Effect of outliers in test data on the comparison of LS and LS-CV methods**

| | n=50 nl= 0.2<br>no outlier | | | n=50, nl= 0.2<br>5 outliers - training | | | n=50, nl= 0.2<br>5 outliers - test | | |
|---|---|---|---|---|---|---|---|---|---|
| | **LS** | **LS-CV** | **%imp.** | **LS** | **LS-CV** | **%imp.** | **LS** | **LS-CV** | **%imp.** |
| **noisy test MSE** | 7.93 | 7.87 | 0.76% | 8.79 | 8.12 | 8% | 26.53 | 26.53 | 0.015% |

Note that addition of outliers to the test dataset will throw off any fitting method, unless data is preprocessed with existing outlier filtering tools[61,62]. A separate run, in which five outliers are added only to the noisy test dataset, shows no significant improvement between LS and LS-CV methods, as shown in Table 8. In this case, the model is trained with comparatively good data and then its "accuracy" is tested again bad data, which is of course inherently doomed since the measure of accuracy is fundamentally flawed. This simple “sanity test” highlights the importance of sampling when using a real-life experimental test dataset for model validation purposes: great care should be taken to assure that any test data is as free of outliers as possible since the calculated accuracy of the fitting may otherwise result in an incorrect interpretation of the results. While here the “flawed” test data (i.e. test data with outliers) was kept artificially constant, CV methods will minimize the danger of this occurring, due to the partition of dataset into folds and repeated estimations based on all permutations of these folds. This assures a sampling of data in a way that is more resistant to the presence of outliers since only a subset of the runs will contain outliers in the test data (as illustrated further above).

## 4. Conclusion

Despite recent advancements in high-throughput screening, availability of kinetic data in industrial practice is often limited and identification of kinetic parameters has to proceed based on a small number of data points. In this work, we conducted a proof-of-concept evaluation of different algorithms for kinetic parameter estimation from limited data, where cross-validation combined with nonlinear least-squares fitting (LS-CV) is evaluated against conventional least-squares (LS), Markov chain Monte Carlo (MCMC), and genetic algorithm (GA) routines.

Our results show that LS is, unsurprisingly, overall the fastest but least accurate method in predicting kinetics. While GA and MCMC are found to be effective for larger data set sizes with lower measurement noise, LS-CV strongly outperforms these methods when the training data is very noisy. In addition, LS-CV requires significantly fewer objective function evaluations compared to stochastic methods, i.e. it is computationally more efficient.

A more detailed comparison between LS-CV and LS confirms that, remarkably, the former can effectively unveil the true kinetics underlying noisy data. Similarly, LS-CV is also successful in filtering out outliers in the experimental data and capturing the true kinetics, due to the use of repeated estimations on different partitions of the training dataset. This randomized partitioning ("folding") inherent to CV also allows the data to be sampled more homogeneously and make use of the entire available data set, which is critical for a more accurate model validation, in particular when few data are available.

Overall, our study indicates that the implementation of a cross-validation routine to a simple nonlinear least-squares fitting algorithm can provide a robust, easy-to-use, and highly efficient approach to identifying reliable reaction kinetics in the face of limited and/or noisy data, and hence constitutes a valuable tool for researchers and practitioners alike.

## References

1. Levenspiel O. Chemical reaction engineering. *Industrial & engineering chemistry research.* 1999;38(11):4140-4143.

2. Davis ME, Davis RJ. *Fundamentals of chemical reaction engineering.* Courier Corporation; 2012.

3. Fogler HS. *Essentials of Chemical Reaction Engineering: Essenti Chemica Reactio Engi.* Pearson Education; 2010.

4. Matera S, Schneider WF, Heyden A, Savara A. Progress in accurate chemical kinetic modeling, simulations, and parameter estimation for heterogeneous catalysis. *ACS Catalysis.* 2019;9(8):6624-6647.

5. Maria G. A review of algorithms and trends in kinetic model identification for chemical and biochemical systems. *Chemical and Biochemical Engineering Quarterly.* 2004;18(3):195-222.

6. Labovský J, Švandová Z, Markoš J, Jelemenský Lu. Mathematical model of a chemical reactor—useful tool for its safety analysis and design. *Chemical Engineering Science.* 2007;62(18-20):4915-4919.

7. Hougen OA, Watson KM. *Chemical process principles.* Vol 1: J. Wiley & Sons, inc.; 1943.

8. Smith JM. *Chemical engineering kinetics.* 1981.

9. Sinnott R. *Chemical engineering design.* Vol 6: Elsevier; 2014.

10. Buzzi-Ferraris G, Manenti F. *Interpolation and regression models for the chemical engineer: Solving numerical problems.* John Wiley & Sons; 2010.

11. Hill CG, Root TW. *An introduction to chemical engineering kinetics & reactor design.* Wiley Online Library; 1977.

12. Venkatasubramanian V. The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal.* 2019;65(2):466-478.

13. Kano M, Nakagawa Y. Data-based process monitoring, process control, and quality improvement: Recent developments and applications in steel industry. *Computers & Chemical Engineering.* 2008;32(1-2):12-24.

14. Martín M, Adams II TA. Challenges and future directions for process and product synthesis and design. *Computers & Chemical Engineering.* 2019;128:421-436.

15. Maria G, Muntean O. Model reduction and kinetic parameters identification for the methanol conversion to olefins. *Chemical engineering science.* 1987;42(6):1451-1460.

16. Park T-Y, Froment GF. Kinetic modeling of the methanol to olefins process. 2. Experimental results, model discrimination, and parameter estimation. *Industrial & engineering chemistry research.* 2001;40(20):4187-4196.

17. Wolberg J. *Data analysis using the method of least squares: extracting the most information from experiments.* Springer Science & Business Media; 2006.

18. Singer AB, Taylor JW, Barton PI, Green WH. Global dynamic optimization for parameter estimation in chemical kinetics. *The Journal of Physical Chemistry A.* 2006;110(3):971-976.

19. Wang F-S, Su T-L, Jang H-J. Hybrid differential evolution for problems of kinetic parameter estimation and dynamic optimization of an ethanol fermentation process. *Industrial & engineering chemistry research.* 2001;40(13):2876-2885.

20. Eftaxias A, Font J, Fortuny A, Fabregat A, Stüber F. Nonlinear kinetic parameter estimation using simulated annealing. *Computers & chemical engineering.* 2002;26(12):1725-1733.

21. Albrecht J. Estimating reaction model parameter uncertainty with Markov Chain Monte Carlo. *Computers & Chemical Engineering.* 2013;48:14-28.

22. Zhang LA, Urbano A, Clermont G, Swigon D, Banerjee I, Parker RS. APT-MCMC, a C++/Python implementation of Markov Chain Monte Carlo for parameter identification. *Computers & chemical engineering.* 2018;110:1-12.

23. Geyer CJ. Practical markov chain monte carlo. *Statistical science.* 1992:473-483.

24. Whitley D. A genetic algorithm tutorial. *Statistics and computing.* 1994;4(2):65-85.

25. Khansary MA, Sani AH. Using genetic algorithm (GA) and particle swarm optimization (PSO) methods for determination of interaction parameters in multicomponent systems of liquid–liquid equilibria. *Fluid Phase Equilibria.* 2014;365:141-145.

26. Mühlenbein H, Schomisch M, Born J. The parallel genetic algorithm as function optimizer. *Parallel computing.* 1991;17(6-7):619-632.

27. Qin SJ, Chiang LH. Advances and opportunities in machine learning for process data analytics. *Computers & Chemical Engineering.* 2019;126:465-473.

28. Qin SJ. Process data analytics in the era of big data. In: Wiley Online Library; 2014.

29. Himmelblau DM. Accounts of experiences in the application of artificial neural networks in chemical engineering. *Industrial & Engineering Chemistry Research.* 2008;47(16):5782-5796.

30. Himmelblau DM. Applications of artificial neural networks in chemical engineering. *Korean journal of chemical engineering.* 2000;17(4):373-392.

31. Bonvin D, Georgakis C, Pantelides C, et al. Linking models and experiments. *Industrial & Engineering Chemistry Research.* 2016;55(25):6891-6903.

32. Hawkins DM. The problem of overfitting. *Journal of chemical information and computer sciences.* 2004;44(1):1-12.

33. McLean KA, McAuley KB. Mathematical modelling of chemical processes—obtaining the best model predictions and parameter estimates using identifiability and estimability procedures. *The Canadian Journal of Chemical Engineering.* 2012;90(2):351-366.

34. Graciano J, Mendoza DF, Le Roux GA. Performance comparison of parameter estimation techniques for unidentifiable models. *Computers & Chemical Engineering.* 2014;64:24-40.

35. Foss BA, Lohmann B, Marquardt W. A field study of the industrial modeling process. 1998.

36. Hawkins DM, Kraker J. Deterministic fallacies and model validation. *Journal of chemometrics.* 2010;24(3‐4):188-193.

37. Tan N, Rao H, Li Z, Li X. Prediction of chemical carcinogenicity by machine learning approaches. *SAR and QSAR in Environmental Research.* 2009;20(1-2):27-75.

38. Martin TM, Harten P, Young DM, et al. Does rational selection of training and test sets improve the outcome of QSAR modeling? *Journal of chemical information and modeling.* 2012;52(10):2570-2578.

39. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. Paper presented at: Ijcai1995.

40. Kulkarni A, Jayaraman VK, Kulkarni BD. Support vector classification with parameter tuning assisted by agent-based technique. *Computers & chemical engineering.* 2004;28(3):311-318.

41. Pirdashti M, Curteanu S, Kamangar MH, Hassim MH, Khatami MA. Artificial neural networks: applications in chemical engineering. *Reviews in Chemical Engineering.* 2013;29(4):205-239.

42. Stone M. Cross‐validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological).* 1974;36(2):111-133.

43. Kim M, Lee Y-H, Han C. Real-time classification of petroleum products using near-infrared spectra. *Computers & Chemical Engineering.* 2000;24(2-7):513-517.

44. Coley CW, Green WH, Jensen KF. Machine learning in computer-aided synthesis planning. *Accounts of chemical research.* 2018;51(5):1281-1289.

45. Coley CW, Barzilay R, Jaakkola TS, Green WH, Jensen KF. Prediction of organic reaction outcomes using machine learning. *ACS central science.* 2017;3(5):434-443.

46. Marquardt DW. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics.* 1963;11(2):431-441.

47. Choi Y, Stenger HG. Water gas shift reaction kinetics and reactor modeling for fuel cell grade hydrogen. *Journal of Power Sources.* 2003;124(2):432-439.

48. Roy A, Clermont G, Daun S, Parker RS. A mathematical model of acute inflammatory response to endotoxin challenge. Paper presented at: AIChE Annual Meeting, Salt Lake City, UT, 538pp2007.

49. Zak DE, Stelling J, Doyle III FJ. Sensitivity analysis of oscillatory (bio) chemical systems. *Computers & chemical engineering.* 2005;29(3):663-673.

50. Kreutz C, Raue A, Kaschek D, Timmer J. Profile likelihood in systems biology. *The FEBS journal.* 2013;280(11):2564-2571.

51. Lilliefors HW. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American statistical Association.* 1967;62(318):399-402.

52. Massey Jr FJ. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association.* 1951;46(253):68-78.

53. Neyman J. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. In: *Breakthroughs in Statistics.* Springer; 1992:123-150.

54. Fushiki T. Estimation of prediction error by using K-fold cross-validation. *Statistics and Computing.* 2011;21(2):137-146.

55. Yadav S, Shukla S. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. Paper presented at: 2016 IEEE 6th International Conference on Advanced Computing (IACC)2016.

56. Hastings WK. Monte Carlo sampling methods using Markov chains and their applications. 1970.

57. Chib S, Greenberg E. Understanding the metropolis-hastings algorithm. *The american statistician.* 1995;49(4):327-335.

58. Van Laarhoven PJ, Aarts EH. Simulated annealing. In: *Simulated annealing: Theory and applications.* Springer; 1987:7-15.

59. Stein M. Large sample properties of simulations using Latin hypercube sampling. *Technometrics.* 1987;29(2):143-151.

60. Raue A, Kreutz C, Maiwald T, et al. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics.* 2009;25(15):1923-1929.

61. Wang H, Bah MJ, Hammad M. Progress in outlier detection techniques: A survey. *IEEE Access.* 2019;7:107964-108000.

62. Zimek A, Filzmoser P. There and back again: Outlier detection between statistical reasoning and data mining algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery.* 2018;8(6):e1280.