Adapting Multicomponent Predictive Systems using Hybrid Adaptation Strategies with Auto-WEKA in Process Industry

Manuel Martin Salvador MSALVADOR@BOURNEMOUTH.AC.UK
Marcin Budka MBUDKA@BOURNEMOUTH.AC.UK
Bogdan Gabrys BGABRYS@BOURNEMOUTH.AC.UK
Data Science Institute. Bournemouth University. Poole, UK

Abstract

Automation of composition and optimisation of multicomponent predictive systems (MCPSs) made of a number of preprocessing steps and predictive models is a challenging problem that has been addressed in recent works. However, one of the current challenges is how to adapt these systems in dynamic environments where data is changing over time. In this work we propose a hybrid approach combining different adaptation strategies with the Bayesian optimisation techniques for parametric, structural and hyperparameter optimisation of entire MCPSs. Experiments comparing different adaptation strategies have been performed on 7 datasets from real chemical production processes. Experimental analysis shows that optimisation of entire MCPSs as a method of adaptation to changing environments is feasible and that hybrid strategies perform better in most of the analysed cases.

Keywords: Adaptive systems; Automatic predictive model building and parametrisation; Multicomponent predictive systems; Chemical production processes; Bayesian optimisation

1. Introduction

Development of data-driven predictive models in the process industry has traditionally been a labour-intensive process, requiring expert knowledge (see e.g. Lin et al. (2007)). Data preprocessing plays a crucial role in building effective models as raw data has many imperfections (e.g. outliers or missing values) and is typically high-dimensional. It usually takes days or even weeks of work to prepare a workflow made of preprocessing and data transformation methods that effectively cleans a dataset (Pyle (1999); Pearson (2005)). One of the challenges that practitioners face nowadays is the composition of workflows which involves choosing among a large number of algorithms and hyperparameters to tackle each individual step of the flow. Recent advances in meta-learning (e.g. Lemke and Gabrys (2010); Jankowski and Grabczewski (2011); Vanschoren (2011); Lemke et al. (2013)), automated planning (e.g. Serban et al. (2012); Fernández et al. (2013)) and Bayesian optimisation (e.g. Hutter et al. (2011); Thornton et al. (2013); Feurer et al. (2015)) have made a great progress in automating this tedious process.

A formal representation of workflows as Multi-Component Predictive Systems (MCPS) was presented in Martin Salvador et al. (2016), where preprocessing methods and learning algorithms are transitions of a Petri net (Petri (1962)). See for example Figure 2 that shows the a hierarchical MCPS. In our previous work, we showed how the composition and optimisation of MCPSs can be automated for a given dataset using approaches like Bayesian optimisation.

Datasets from chemical production processes contain readings from physical sensors that are located in different parts of the chemical plants. These sensors measure values such as temperatures, flows and pressures that are constantly changing during chemical reactions. Some reactions are quite stable and data distribution does not change over time. Others, however, can vary significantly between production batches or even within a single, long-running production process (e.g. Sharmin et al. (2006)). In addition, the degradation of sensors over long periods of time produce a change in the input values that can severely affect predictive performance.

There are different predictive models adaptation strategies for dealing with such changing environments (see e.g. Kadlec et al. (2011) for a review of various adaptation mechanisms in the process industry context or Gama et al. (2014) for a more general survey). For example, active detection techniques monitor a certain measure over time and react when its running average changes significantly or goes over a given threshold. These techniques require optimisation of parameters like the threshold value or averaging period, and can also lead to false positives. On the other hand, passive adaptation techniques update the model periodically with new data (e.g. most recent data or most representative samples), even if it is not strictly necessary (Žliobaitė et al. (2015)).

Although we have quite extensively investigated and proposed a number of very flexible solutions to the problem of adapting predictive models in previous works (e.g. Kadlec and Gabrys (2009), Kadlec and Gabrys (2011)), Bakirov et al. (2015), Žliobaitė et al. (2015)), in this paper we explore the feasibility and effectiveness of deploying a parametric, structural and hyperparameter optimisation of a complete MCPSs in chemical production processes. In this type of processes, predictions are delivered online but the ground truth is delayed and only available in batches. To this end, 7 datasets from real chemical processes are used to compare four MCPS adaptation strategies – two of them including a Sequential Model-Based Optimization method (SMBO) – against a static approach used as a baseline.

1.1. CASH problem for MCPS

A multicomponent predictive system can be represented as a WA-WF-net (Well-handled and Acyclic Workflow Petri net, Ping et al. (2004); Van Der Aalst (1998)). Formally,

$$MCPS = (P, T, F) \tag{1}$$

is a directed acyclic graph where P and T are finite sets of nodes called places and transitions, respectively, and F are the arcs connecting nodes. In an MCPS, a place can contain a single token which is represented as a tensor (i.e. multidimensional array). Further definition and properties of an MCPS are presented in Martin Salvador et al. (2016).

The Combined Algorithm Selection and Hyperparameter configuration (CASH) problem originally presented by Thornton et al. (2013) and then extended in Martin Salvador et al. (2016) consists of finding the best combination of algorithms and hyperparameters forming an $MCPS = (P, T_{\lambda^*}, F)^*$ that optimises an objective function $\mathcal{L}$ (e.g. Equation 2 minimises the k-fold cross-validation error) for a given dataset $\mathcal{D}$.

$$(P, T_{\lambda^*}, F)^* = \underset{(P,T,F)^{(j)} \in \mathcal{A}, \lambda \in \Lambda^{(j)}}{\arg\min} \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}((P, T_\lambda, F)^{(j)}, \mathcal{D}^{(i)}_{train}, \mathcal{D}^{(i)}_{valid}) \tag{2}$$

where $\mathcal{A} = \{A^{(1)}, \dots, A^{(k)}\}$ is a set of MCPSs with associated hyperparameter spaces $\Lambda^{(1)}, \dots, \Lambda^{(k)}$. The loss function $\mathcal{L}$ takes as arguments an algorithm configuration A_λ (i.e. an instance of an MCPS configuration and its hyperparameters), a training set $\mathcal{D}^{(i)}_{train}$ and a validation set $\mathcal{D}^{(i)}_{valid} = \mathcal{D} \setminus \mathcal{D}^{(i)}_{train}$.

The CASH problem as defined above has been addressed in our previous work (Martin Salvador et al. (2016)) using SMBO methods in which we were able to effectively build MCPSs for a number of datasets, including those that we use for experimentation in this paper. However, to the best of our knowledge, the problem of adapting the deployed MCPSs using SMBO methods has not been approached yet in the literature.

2. Adaptation of MCPS in process industry

2.1. Datasets

The datasets used in these experiments are listed in Table 1 and have been extensively used in the literature (see e.g. Fortuna et al. (2003, 2005); Kadlec and Gabrys (2009, 2011); Budka et al. (2014); Bakirov et al. (2015); Martin Salvador et al. (2016)). They contain instances made of sensor readings from different chemical production processes and the state of the target value to be predicted (i.e. low, normal, high). In the case of processes where two products are measured, there are 9 classes representing a combination of the 3 states of each product output. The initial 70% of instances were used for training and tuning of the models while the subsequent 30% were reserved for testing. The test set has been split into 10 batches of approximately equal size.

			Instances		
Name	**Attributes**	**Classes**	**Total**	**Initial training**	**Batch**
absorber	38	3	1599	1119	48
catalyst	14	3	5867	4109	176
debutanizer	7	3	2394	1676	72
drier	19	3	1219	853	37
oxeno	71	3	17588	12311	528
sulfur	5	9	10081	7057	303
thermalox	38	9	2820	1974	58

Table 1: Datasets properties

2.2. Adaptation strategies

Four different adaptation strategies have been selected for a comparison within the same evaluation framework. All these approaches assume that an MCPS has been composed and optimised using SMAC (Sequential Model-based Algorithm Configuration Hutter et al. (2011)) integrated in our Auto-WEKA extension (Martin Salvador et al. (2016)) during 30 CPU-hours for the initial training set. SMAC incrementally builds a random forest to model

Strategy	Data for Training	Forgetting	Parametric Adaptation	MCPS Optimisation
Baseline	No	No	No	No
Batch	Batch	Yes	Yes	No
B + SMAC	Batch	Yes	Yes	Yes
Cumulative	Cumulative	No	Yes	No
C + SMAC	Cumulative	No	Yes	Yes

Table 2: Evaluated strategies

the relation between hyperparameters and predictive performance by exploring promising configurations in a search space. The resultant MCPS is then used to predict the target value of a batch of incoming instances from the test set. After that, the true labels of the batch are provided and one of the following strategies is executed (for summary of the strategies please refer to Table 2):

- **Baseline** does not make adaptation of any kind. The initial MCPS continues predicting the labels for the consecutive batches.
- **Batch**, where a new MCPS is trained using only the labeled data from the most recent batch and the configuration (including hyper-parameters) from the initial MCPS. This strategy learns new concepts and forgets the old ones.
- **Batch + SMAC**, where a new MCPS is composed using the most recent batch as training set and SMAC as optimisation strategy for 5 CPU-hours (see Figure). Although the old concepts are being forgotten, the historical information of the underlying SMAC model remains. That is, the random forest built by SMAC (made of runs and their classification errors) is preserved, so the exploration won't start from scratch.
- **Cumulative**, where a new MCPS is trained using all the available labelled instances including the most recent batch, but the configuration of the initial MCPS is preserved. Therefore, there is no forgetting of concepts.
- **Cumulative + SMAC** uses the same training strategy as Cumulative, but similarly to Batch+SMAC strategy a new MCPS is composed using SMAC after every batch for 5 CPU-hours. The historical information of the underlying SMAC model remains.

Each experiment has been repeated 25 times with different random initialisations of SMAC (i.e. seeds), but keeping the same data partitions. Results in Section 3 are therefore aggregated over the 25 runs.

2.3. Search space

The SMAC search space has been defined to support MCPSs with up to five preprocessing steps, a predictive model and a meta-predictor (1564 possible hyperparameters in total). The nodes of the Petri net are connected in the following order: $i \rightarrow$ missing value handling

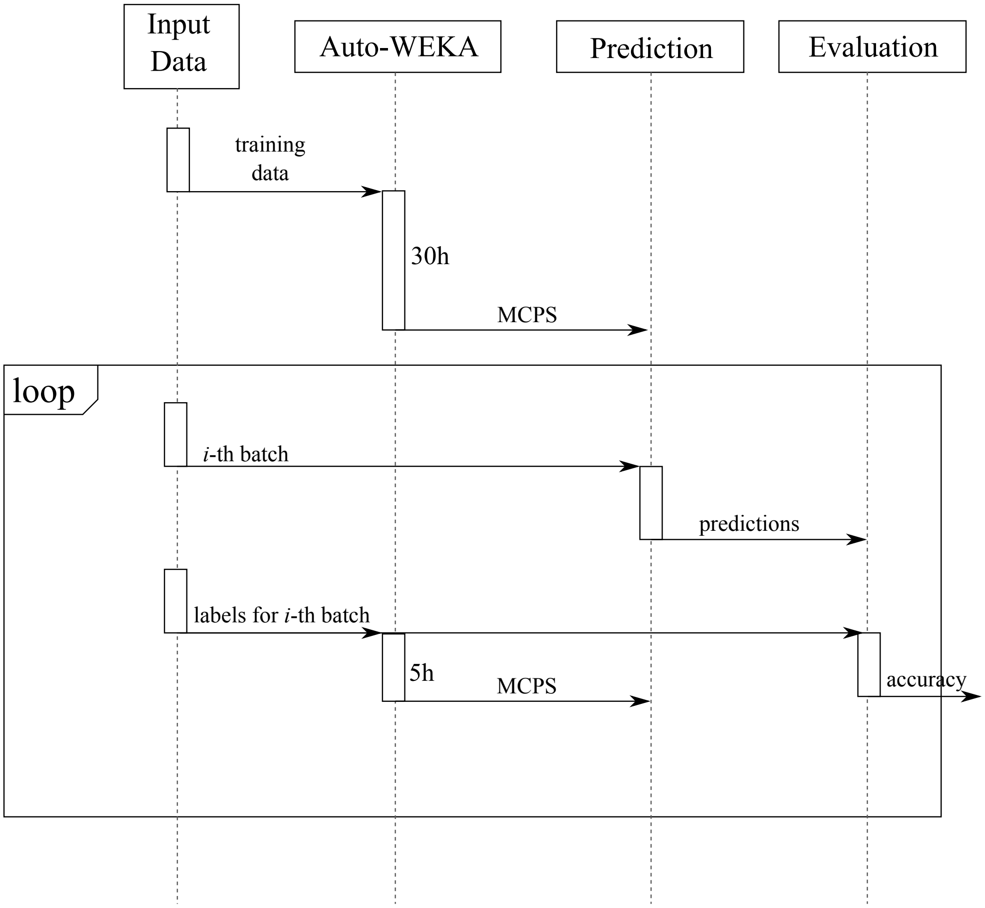


Figure 1: Sequence diagram for Batch+SMAC strategy.

→ p_1 → outlier detection and handling[1] → p_2 → data transformation → p_3 → dimensionality reduction → p_4 → sampling → p_5 → predictor → p_6 → meta-predictor → o, where $p \in P$, i and o are the input and output places, respectively. This arrangement of nodes is based on our experience with the process industry (e.g. Budka et al. (2014)), but the same preprocessing steps are also common in other fields. An example of MCPS following this flow is shown in Figure 2. To find all the WEKA methods that can be selected for each node, please refer to Tables 2 and 3 of Martin Salvador et al. (2016).

3. Results

3.1. Predictive performance

The average classification errors for each dataset and strategy are shown in Table 3, with the last row denoting the average rank (1: best – 5: worst) of each strategy. Plots comparing

[1] Outliers are handled in a different way than missing values

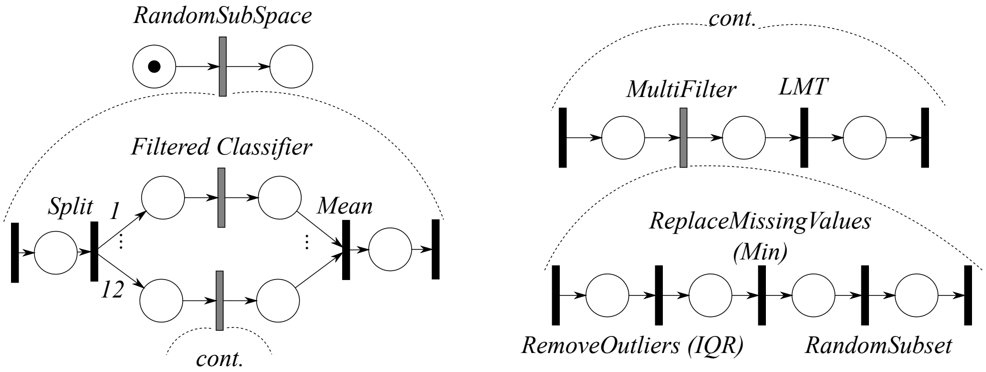


Figure 2: The best MCPS for 'catalyst' dataset in batch 7. The WEKA methods used are explained in Tables 2 and 3 of Martin Salvador et al. (2016).

the batch predictive performance between strategies do not fit in this paper, but can be found in our repository[2].

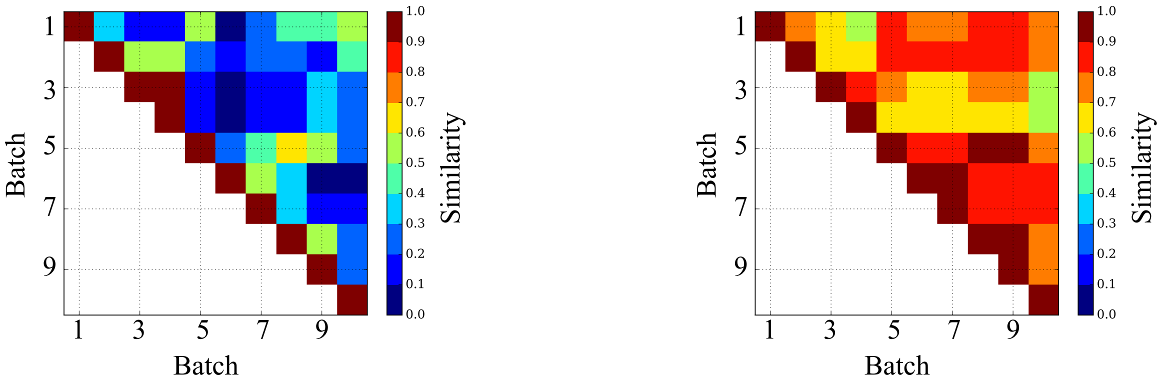
The batch adaptation strategy performed, in average, better than the baseline in 6 out of 7 datasets (i.e. lower mean error). The only dataset in which the predictive accuracy has worsened is 'drier'. This dataset does not change much in terms of any predictable trends and the performance improves with adding and using more data for training the predictor. Therefore one of the main causes of such deterioration is the small batch size used for training (only 37 samples per batch) in comparison to the size of the training data for the baseline method. The use of SMAC with batch strategy has resulted in better performances in 3 out of 7 datasets. That means that over-optimising an MCPS may not always be the best approach when there is a risk of over-fitting due to a drastic forgetting mechanism employed.

Cumulative strategy has improved the predictions for all the datasets. These results were expected to happen since there is no forgetting of previous samples. In addition, applying SMAC optimisation has helped to refine MCPSs and has improved the results of standalone cumulative strategy in 5 out of 7 datasets.

	Base.	**Batch**		**B+SMAC**		**Cumulative**		**C+SMAC**	
	ϵ	ϵ	δ	ϵ	δ	ϵ	δ	ϵ	δ
absorber	54.32	40.13	+14.19	43.43	+10.88	33.37	+20.95	**33.13**	+21.18
catalyst	68.34	25.09	+43.24	**25.06**	+43.27	37.93	+30.41	38.08	+30.25
debutanizer	58.88	**47.54**	+11.34	48.73	+10.15	53.35	+5.53	52.77	+6.11
drier	49.89	55.54	-5.65	54.18	-4.29	**48.12**	+1.77	49.56	+0.33
oxeno	45.92	40.60	+5.33	**38.08**	+7.84	39.44	+6.48	38.70	+7.22
sulfur	80.67	79.91	+0.76	80.19	+0.48	79.70	+0.97	**78.92**	+1.75
thermalox	55.07	39.42	+15.65	35.83	+19.24	39.95	+15.12	**33.25**	+21.81
avg. rank	4.71	3.00		2.29		2.71		2.00	

Table 3: Average % classification error (ϵ) and improvement with respect to baseline (δ) for each dataset and adaptation strategy. Best result of each dataset is in bold.

[2] https://github.com/dsibournemouth/autoweka



a) Batch+SMAC

b) Cumulative+SMAC

Figure 3: MCPS similarity between batches for ‘catalyst’ dataset (seed=17)

3.2. Evolution of MCPS over batches

Analysing the evolution of MCPSs found after applying SMAC optimisation between batches can help to identify how robust they are to changes in data. To calculate the similarity between MCPSs we use a weighted sum of Hamming distances given by

$$d(F,G) = 1 - \frac{\sum_{i=1}^{N} (w_i \cdot \delta_i)}{\sum_{i=1}^{N} w_i} \tag{3}$$

where F and G are MCPS with N transitions, w_i is the weight for the ith transition and δ_i is the Hamming distance (a standard measure of string dissimilarity) at position i. Weights for this particular search space have been set to $W = \{1,1,1,1,1,2,1.5\}$. That is, preprocessing transitions have the same weight while both predictors and meta-predictors have higher weights because of their importance (Hoos et al. (2014)).

Although there is not enough space to include all the results in this paper, we would like to highlight a common pattern that we found. Figure 3 shows two triangular matrices representing the MCPS similarity between batches for ‘catalyst’ dataset in a) Batch+SMAC and b) Cumulative+SMAC strategies. We can observe how MCPSs between batches 3 and 4 are very similar but then there are big change between others. The large differences between the MCPSs in a) are due to the extreme forgetting mechanism that is not present in b), where the MCPSs are more stable due to the accumulation of historical data as no forgetting is used. Table 4 shows the evolution of the MCPS configuration between batches for the same case as Figure 3-b. The meta-predictor is the same for all batches (AdaBoost), varying only its hyperparameters. The logistic model tree (LMT) classifier has been selected in 8 of 10 batches, while a multilayer perceptron (MLP) was chosen for batches 3 and 4. The transformation component is the one presenting more variation. Finally, the method for replacing missing values has slightly varied across the batches.

#	missing values	transformation	predictor	meta-predictor
1	EMImputation -E 346.813 -N 467 -Q 799.631	Standardize	LMT -R -C -P -M 1 -W 0.029 -A	AdaBoostM1 -P 100 -I 60 -Q -S 1
2	CRMV -M 2	Normalize -S 1.0 -T 0.0	LMT -M 15 -W 0	AdaBoostM1 -P 100 -I 35 -Q -S 1
3	CRMV -M 2	Standardize	MLP -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a	AdaBoostM1 -P 100 -I 85 -S 1
4	CRMV -M 2	Center	MLP -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a	AdaBoostM1 -P 100 -I 81 -S 1
5	CRMV -M 2	Standardize	LMT -M 15 -W 0	AdaBoostM1 -P 100 -I 28 -S 1
6	CRMV -M 2	Wavelet	LMT -M 15 -W 0	AdaBoostM1 -P 100 -I 29 -S 1
7	CRMV -M 2	Wavelet	LMT -M 15 -W 0	AdaBoostM1 -P 100 -I 10 -S 1
8	CRMV -M 5	Standardize	LMT -P -M 13 -W 0	AdaBoostM1 -P 100 -I 17 -S 1
9	CRMV -M 1	Standardize	LMT -R -C -M 19 -W 0 -A	AdaBoostM1 -P 95 -I 2 -S 1
10	-	Wavelet	LMT -M 15 -W 0	AdaBoostM1 -P 100 -I 10 -S 1

Table 4: Components found for 'catalyst' dataset for each batch in C+SMAC strategy (seed=17). No outlier handling, dimensionality reduction and sampling components were selected by SMAC for this particular case.

4. Conclusion

This paper introduces a hybrid approach for adapting MCPSs in a chemical processes predictive modelling deployment by combining different adaptation strategies with Bayesian optimisation techniques. An intensive experimental evaluation comparing 5 different strategies using chemical production datasets has shown that such approach got better results for 5 out of 7 datasets. The best strategy has been a combination of cumulative training with SMAC optimisation, highlighting the fact that having more data usually helps but also refining the MCPS makes a considerable improvement in the results.

In future work, we would like to investigate adaptation mechanisms for SMBO methods and the impact they might have in finding better MCPSs.

Acknowledgments

Research leading to these results has received funding from the EC within the Marie Curie Industry and Academia Partnerships and Pathways (IAPP) programme under grant agreement no. 251617. We would like to thank Evonik Industries AG for providing the datasets and support on pursuing this research.

References

R. Bakirov, B. Gabrys, and D. Fay. On sequences of different adaptive mechanisms in non-stationary regression problems. In *IJCNN 2015*, pages 1–8, 2015.

M. Budka, M. Eastwood, B. Gabrys, P. Kadlec, M. Martin Salvador, S. Schwan, A. Tsakonas, and I. Žliobaitė. From Sensor Readings to Predictions: On the Process of Developing Practical Soft Sensors. In *IDA 2014*, volume 8819, pages 49–60, 2014.

S. Fernández, T. de la Rosa, F. Fernández, R. Suárez, J. Ortiz, D. Borrajo, and D. Manzano. Using automated planning for improving data mining processes. *Knowl. Eng. Rev.*, 28 (02):157–173, 2013.

M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter. Efficient and Robust Automated Machine Learning. *NIPS 2015*, pages 2944–2952, 2015.

L. Fortuna, A. Rizzo, M. Sinatra, and M. G. Xibilia. Soft analyzers for a sulfur recovery unit. *Control Eng. Pract.*, 11:1491–1500, 2003.

L. Fortuna, S. Graziani, and M. G. Xibilia. Soft sensors for product quality monitoring in debutanizer distillation columns. *Control Eng. Pract.*, 13:499–508, 2005.

J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4):1–37, 2014.

H. Hoos, U. B. C. Ca, K. Leyton-Brown, and F. Hutter. An Efficient Approach for Assessing Hyperparameter Importance. *ICML 2014*, 32(1):754–762, 2014.

F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential Model-Based Optimization for General Algorithm Configuration. *Learn. Intell. Optim.*, 6683:507–523, 2011.

N. Jankowski and K. Grabczewski. Universal Meta-Learning Architecture and Algorithms. In *Meta-Learning Comput. Intell.*, pages 1–76. 2011.

P. Kadlec and B. Gabrys. Architecture for development of adaptive on-line prediction models. *Memetic Comput.*, 1(4):241–269, 2009.

P. Kadlec and B. Gabrys. Local learning-based adaptive soft sensor for catalyst activation prediction. *AIChE J.*, 57(5):1288–1301, 2011.

P. Kadlec, R. Grbić, and B. Gabrys. Review of adaptation mechanisms for data-driven soft sensors. *Comput. Chem. Eng.*, 35(1):1–24, 2011.

C. Lemke and B. Gabrys. Meta-learning for time series forecasting and forecast combination. *Neurocomputing*, 73(10-12):2006–2016, 2010.

C. Lemke, M. Budka, and B. Gabrys. Metalearning: a survey of trends and technologies. *Artificial Intelligence Review*, pages 1–14, 2013.

B. Lin, B. Recke, J. K.H. Knudsen, and S. B. Jørgensen. A systematic approach for soft sensor development. *Comput. Chem. Eng.*, 31(5-6):419–425, 2007.

M. Martin Salvador, M. Budka, and B. Gabrys. Automatic composition and optimisation of multicomponent predictive systems. *IEEE Transactions on Neural Networks and Learning Systems (under review - available at* `http://bit.ly/automatic-mcps-tnnls`*)*, 2016.

R.K. Pearson. Mining imperfect data. *Soc. Ind. Appl. Mech. USA*, 2005.

C. A. Petri. *Kommunikation mit Automaten.* PhD thesis, Universität Hamburg, 1962.

L. Ping, H. Hao, and L. Jian. On 1-soundness and Soundness of Workflow Nets. In *Third Work. Model. Objects, Components, Agents*, pages 21–36, 2004.

D. Pyle. *Data preparation for data mining.* Morgan Kaufmann, 1999.

F. Serban, A. Bernstein, and S. Fischer. Designing KDD-Workflows via HTN-Planning for Intelligent Discovery Assistance. In *Proc. Int. Work. Plan. to Learn*, pages 10–17, 2012.

R. Sharmin, U. Sundararaj, S. Shah, L. Vande Griend, and Y. Sun. Inferential sensors for estimation of polymer quality parameters: Industrial application of a PLS-based soft sensor for a LDPE plant. *Chem. Eng. Sci.*, 61(19):6372–6384, 2006.

C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pages 847–855, 2013.

W. M. P. Van Der Aalst. The Application of Petri Nets To Workflow Management. *J. Circuits, Syst. Comput.*, 08(01):21–66, 1998.

J. Vanschoren. Meta-learning architectures: Collecting, organizing and exploiting meta-knowledge. *Stud. Comput. Intell.*, 358:117–155, 2011.

I. Žliobaitė, M. Budka, and F. Stahl. Towards cost-sensitive adaptation: When is it worth updating your predictive model? *Neurocomputing*, 150, Part A:240–249, 2015.