Manuscript version
DOI original paper: 10.1109/ICARCE55724.2022.10046584

Development of a Framework for Data-Driven Modeling with Cloud Services in the Process Industry

Dominik Polke*, Florian Diepers, Elmar Ahle
Electrical Engineering and Computer Science
University of Applied Sciences Niederrhein
Krefeld, Germany
{dominik.polke, florian.diepers, elmar.ahle}@hs-niederrhein.de

Dirk Söffker
Dynamics and Control
University of Duisburg-Essen
Duisburg, Germany
soeffker@uni-due.de

***Abstract*—The chemical process industry is currently undergoing a transformation to Chemistry 4.0, where digitalization, modularization, sustainability, and the circular economy are coming into focus. A growing interest in the use of process data with the aim of gaining a better understanding of the production process and conserving resources can be observed. Data-driven modeling is used in chemical industry when the production process is too complex to be described by chemical laws. Gaining knowledge of the chemical relationships can lead to resource-conserving production. In this paper, a framework to optimize the process of data-driven modeling in an industrial environment is presented. For generating data-driven models of industrial processes, many manual and time-consuming steps have to be carried out. This leads to delay in information acquisition and process optimization. Therefore, the presented framework automates these steps to accelerate the process of data-driven modeling. The steps are to extract the data from a process control system (PCS), make the data available for data-driven modeling, train the model, and deploy the model for predicting the process. To achieve high availability of the data and generate data-driven models, cloud services are used. The framework of this paper is applied to a high-throughput formulation system (HTFS) for coatings. In this paper, Gaussian processes are used for data-driven modeling. The evaluation of the framework shows the usefulness in this domain, but also the flexibility and scalability of this framework.**

Keywords*— *data-driven modeling, high-throughput coating formulation, cloud computing, Gaussian process, Chemistry 4.0, process industry

I. Introduction

Chemistry 4.0 is the new era of chemical process industry, in which digitalization, sustainability, and circular economy play key roles [1]. This includes the use of artificial intelligence to learn from experimental data, to gain a better understanding of the production processes, and to conserve resources [2].

In the process industry, the PCS is the data aggregation point of an industrial plant. There are different types of data in industrial processes, like numeric representations of parameter and process values as well as images [3]. The variety of data poses a huge difficulty to integrate, access, and query data in an efficient way. Data lakes have been proposed as a solution to this problem [4]. They provide a common access interface and are repositories storing the raw data in their original formats. This leads to reducing the upfront integration costs and provides more flexibility in data integration [5].

One aim of chemical engineering is to develop new materials. This is mostly based on empirical experience due the complexity of the chemical processes. It is possible to model certain properties of a real coating by chemical laws, but it often fails due to the complexity of chemical composition. Therefore, data-driven modeling is used [6]. In [7], the usage of data-driven modeling for coating technologies is shown with a combination of neural networks and Gaussian processes. The batchwise and adaptive design of experiment (DoE) increases the efficiency of the chemical formulation and reduces the number of experiments necessary compared to classical approaches [7]. Reducing the number of experiments, leads to reduction of resources and costs for production [8].

In [3], the whole life cycle of a machine learning (ML) model is considered for the process industry in general. In this work, the data-preprocessing, learning, and deployment is adapted to a framework in an automated way. It is implemented in the field of coating formulation. The implementation is done and evaluated at a HTFS at the Institute of Surface Technology of the University of Applied Sciences Niederrhein, Krefeld, Germany (HIT).

The process to get the necessary data from an industrial plant and the data-driven modeling with new incoming data are not automated in the HTFS and consume time. Data collection with more stages such as data cleaning and data labeling, has always been the bottleneck for data-driven modeling approaches [3]. To store data and compute data-driven models in a flexible, scalable, and high performing way, cloud computing services are used [9].

The contribution of this paper is the automation of the data-driven learning process in an industrial environment. This paper is structured as follows. First, a short introduction and further references to an existing architecture, cloud computing, data lake, and the usage of Gaussian processes are given in section II. In section III, the automation framework is presented. The framework is implemented in section IV and the evaluation is presented in section V. Finally, a short summary and an outlook of future work are given.

This work has been sponsored by the German Federal Ministry of Education and Research in the funding program "Forschung an Fachhochschulen", project I^{2}DACH (grant no. 13FH557KX0, https://www.hs-niederrhein.de/i2dach).

II. Background and Related Work

In this section, the necessary components to develop a framework for data-driven modeling with cloud services in the process industry are presented. These consist of a general architecture, cloud services, data lake, and a modeling algorithm.

A. NAMUR Open Architecture

In the process industry an architecture is developed to extend the traditional and hierarchical automation structure, which is called NAMUR Open Architecture (NOA) [10]. For faster access and data exchange between the layers of the existing automation structure, NOA provides an open and secure side-channel. The guideline for the implementation of NOA is currently under development. The first three guideline sheets are available, containing the general concept [10], the information model [11], and the security gateway [12]. In the concept of NOA, the existing automation structure is called Core Process Control (CPC). The NOA aims to make production data of the CPC easy and secure usable for Monitoring and Optimization (M+O) [10].

The concept of NOA is visualized in Fig.1 and includes different modules which are defined as follows:

- NOA Information Model: Defines the syntax and semantics of data.
- NOA Diode and IT-Security: Defines the unidirectional data flow from the CPC domain to the M+O domain.
- NOA Verification of Request: Defines the safe and reliable return path of data from the M+O domain to the CPC domain.
- NOA Aggregating Server: Defines and structurizes the data communication paths from the various NOA diodes and M+O applications.

Information are sent to the plant specific M+O via the NOA Diode, which is pointing the information flow. The plant specific M+O collects all data. For the use of advanced analytics like predictive maintenance [13] or data-driven modeling, an additional channel to the central M+O is provided.

B. Cloud Service Platform

Cloud computing can be seen as a service-oriented architecture, which includes almost all computing capabilities [14]. Many cloud computing platforms exist, but with Microsoft Azure, Amazon AWS, and Google Cloud Platform, three platforms exist, which rule the cloud computing market [9].

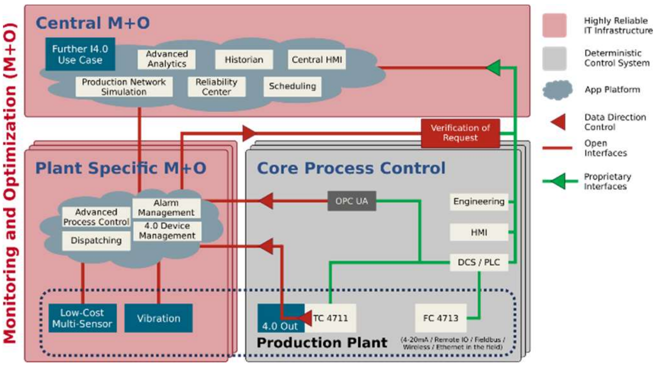


Fig. 1: NAMUR Open Architecture overview (acc. to [10])

These three cloud computing platforms are compared to each other in several papers for different use cases, like in [9], [14], and [15]. The performance of Microsoft Azure and Amazon AWS is above Google Cloud Platform in almost every case. The comparisons of different cloud service aspects, such as computing or storage performance, show that Amazon AWS and Microsoft Azure are in a head-to-head race. According to [16], Microsoft Azure is overall more cost effective. Therefore, Microsoft Azure is used in this work.

C. Data Lake

In [4], a data lake is defined as a flexible, scalable data storage, and management system. This kind of storage contains raw data from heterogeneous sources in their original format. It also allows users to query and explore data. Data lakes do not need any information about the schematic or mappings of the stored data. But, without any management of metadata, a data lake can turn into a so called data swamp. Therefore, according to [4], it is important to extract as much metadata as possible from the data sources.

D. Data-Driven Modeling

In this work, Gaussian processes (GPs) are used for data-driven modeling. The GPs are a probabilistic approach using kernel machines [17]. The principle advantage over other kernel-based approaches is given by a measurement, how certain predictions are. In [7], it was shown, that GPs work well in the process industry.

A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution. The mean function $m(x)$ and the covariance function $k(x, x')$ (also called kernel function) are completely specifying the GP. The GP can be written as

$$f(x) \sim \mathrm{GP}\big(m(x), k(x, x')\big). \tag{1}$$

The development of coatings often includes the usage of many different chemicals. This leads to a high dimensional feature space for the algorithm. For distance-based machine learning algorithms, this leads to the curse of dimensionality [18]. To avoid this, unimportant features can be removed with automatic relevance determination (ARD) by techniques, which are presented in [19]. For GPs the ARD is realized by introducing a lengthscale parameter for each feature [20], which has been used successful in [21].

In this work, the squared exponential kernel [22] is used for modeling. It is defined by

$$k(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{l^2}\right). \tag{2}$$

With the lengthscale l and the output variance σ two hyperparameters are used for fitting the model to the given data. For fitting the model, the maximum likelihood estimation (MLE)

$$\begin{aligned}\ln\big(p(y|\mu, \sigma^2)\big) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \mu)^2 - \frac{N}{2}\ln(\sigma^2) \\ - \frac{N}{2}\ln(2\pi)\end{aligned} \tag{3}$$

is used. The Adam optimizer [23] as a gradient descent method is used in this paper to search for the optimal hyperparameters of the GP.

III. Automation Framework for Data-Driven Modeling

The main goal of the framework, presented in this paper, is to automate the workflow from generating a DoE to train and deploy a ML model. The architecture describes the static components of the framework that interact with each other. The workflow includes the execution of these architecture components.

A. *Architecture of the Framework*

The currently manually executed processes such as data extraction and ML model training need to be automated, because the work carried out by humans is cumbersome and can lead to errors. In case of errors in handling data, the ML model will have no practical relevance. Another reason to automate these processes is the reduction of computational load in comparison to manual data processing and model training.

In addition, the data of the DoE are uniquely assigned, whereby data consistency is given in the database of the PCS. With the automated extraction from the local database of the PCS to the cloud, data have higher availability. Because of this, it is possible for any authorized user to access the process data location-independent, to analyze the process in different ways, without many manual steps.

In Fig. 2, the automation framework is shown. It is based on the idea of NOA. The PCS and the industrial process plant are parts of the CPC. The data are generated in the industrial process plant. These data are composed of parameters and characterization variables. The parameters can be further subdivided into chemical composition and process parameters.

The data are acquired in the PCS and stored in a proprietary database as well as transferred to the cloud via the cloud gateway. Here, the cloud gateway is the bridge between the proprietary database and the cloud. In the context of NOA, the cloud gateway can be seen as the NOA aggregating server as a part of the plant specific M+O. This includes the firewall, encryption and authentication.

In this architecture, as in the NOA concept, only automated data traffic from the CPC domain to the M+O domain is allowed, to prevent uncontrolled feedback to the PCS. The cloud includes the data storage and computational power to train the model. For storing the process data, a data lake is used. In addition, the scripts to be executed for data pre-processing and model training are located in the data lake.

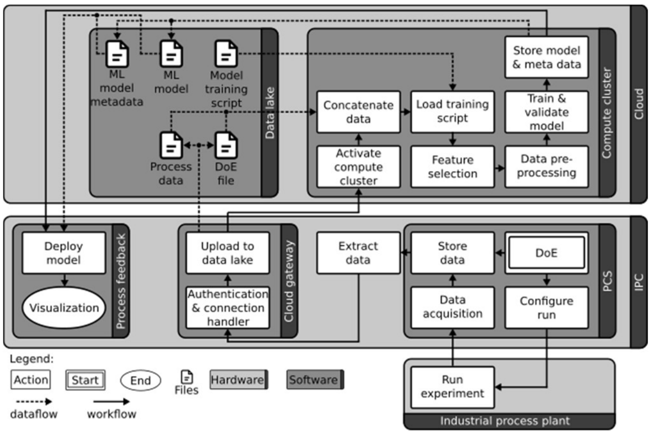


Fig. 2: Automation Framework of data-driven modeling

The model training takes place in compute clusters, which are scalable for low or high computing tasks, depending on the ML tasks, like regression or image classification. The outsourcing of the model training in the cloud is needed for scalability.

B. *Workflow of the Framework*

In the following, the workflow of the framework is presented, which is also shown in Fig. 2. First, the workflow starts with the DoE, which is done by a chemical engineer. At the start of the DoE, a unique identifier (ID) is assigned to the new product being developed. With this ID, the data from the DoE and the measured data of the process plant can be uniquely assigned. The chemist first determines the initial experiments and which parameters are held constant over the series of experiments and which are varied. This information is labeled in the DoE file. Furthermore, the target value for the ML algorithm is defined in this file. Finishing the DoE, the file is uploaded to the data lake in the cloud.

It is assumed, that the DoE file is read into the PCS, which configures the run and the run is carried out on the industrial process plant. A run contains several experiments. In the production of coatings, an experiment is a single chemical formulation. After finishing the run, the process data are recorded and stored in a proprietary database.

After that, the data are extracted from the database and uploaded to the data lake in the cloud via the cloud gateway. If there is no existing folder for the new product, it will be generated. In the case of an existing folder, the data from the new batch is concatenated with the previous batches. After activating the compute cluster, a script is searching for the ID of the product in the data lake.

The compute cluster loads and executes the ML training script. This script is used to select the variable features and the target variable from the data for the ML algorithm. The model training follows with the search of hyperparameters, which fit best to the training data. The last steps are validating the model and storing the model with its metadata in the data lake.

Finally, the model can be deployed and used for predictions. The results are visualized for the chemist, who gets a feedback of the product and can plan further experiments.

IV. Application of the Framework in Hightroughput Coating Formulation

This section shows the application of the automation framework and is illustrated in Fig. 3. During a workference at the University of Applied Sciences Niederrhein, data were generated at the HIT. The aim of this workference was to use ML to develop a coating formulation with as few experiments as possible. The coating should be optimized in terms of gloss, haze and hiding power. This means, that the gloss and haze should be minimized and the hiding power should be maximized. The creation of the different coating compositions was carried out in five iterations with an initial set of 15 experiments and four further iterations with ten experiments in each iteration. In total, there is a dataset of 55 experiments available, which were carried out at a HTFS. This dataset contains 17 parameters as inputs and three characterization variables as outputs. Eleven of these input parameters were variable and the remaining six were either dependent or constant.

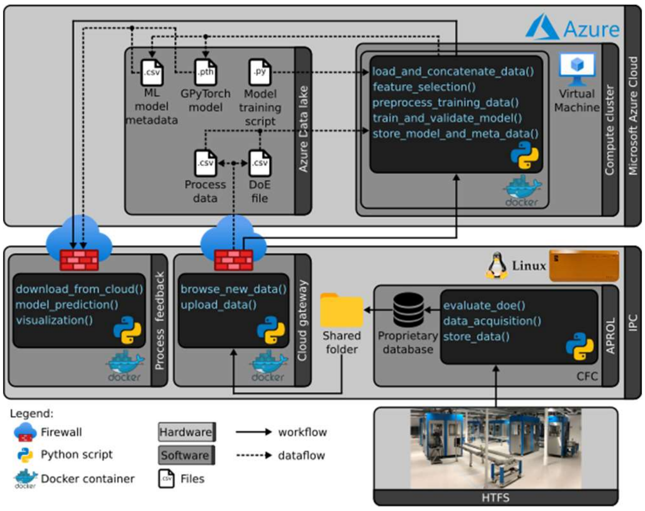


Fig. 3: Application of the framework

These previously recorded data are used and in the following, treated as if just created during testing the workflow. The data are downloaded from the HTFS and stored locally on the IPC to test the workflow. Hence, no direct communication to the HTFS is necessary during the tests. The use of these data is an example to illustrate the automated data flow and the automated process of data-driven modeling.

The implementation starts with the evaluation of the DoE file, which specifies the input and output parameters for the entire workflow. The DoE and the configuration of the run are not part of the implementation in this work, since previously recorded data are used. Apart from this, the entire workflow from section III is implemented with the exemplary data and described in the following.

A. Process Control System

The linux-based PCS APROL from B&R is used in this work. It provides a tamper-proof database as required in areas of the process industry, for example in the pharmaceutical industry. The data acquisition and the saving of the data is executed in the process control system via the Continuous Function Chart (CFC) and an integrated Python script. In this work, the HTFS data are read from the local filesystem as csv files. The DoE and the HTFS data are generated batchwise as database entries. The DoE contains the metadata of the experiments, which are needed to associate DoE data, process data and ML models. Table I shows the exemplary structure of the DoE file. In this table, the first five rows represent metadata. The following rows represent an experiment per row.

TABLE I STRUCTURE OF THE DOE FILE

Date	Time	Author	Prod. ID	Run ID	Num. Exp.
29.07.22	08:54	DP	30	1	10
Parameter$_1$	…	Parameter$_{17}$	Char. var.$_1$	…	Char. var.$_3$
Input	…	Input	Output	…	Output
Var./Const./Dep.	…	Var./Const./Dep.			
Value	Value	Value	Value	Value	Value
⋮	⋮	⋮	⋮	⋮	⋮

B. Cloud Gateway

The PCS has implemented library modules with a Python 2.7 interpreter. The PCS can therefore not access the latest Python libraries, as is necessary for cloud services. Therefore, the cloud gateway is implemented as a Docker container, to get a separated system. Inside of this separated system, the choice of the Python interpreter and the libraries to be used are flexible. Another advantage of choosing a container is, that the PCS stops the container in the event of to high computing power, to not endanger the CPC domain. For the usage of the Azure cloud services, a Python 3.7 interpreter and some libraries of the data lake and machine learning environment are needed. A slim base image with the required Python interpreter and Azure libraries is used in this work. The container and the host system share an assigned folder structure for the exchange of data. The container gets the python scripts from this folder to execute the functions of the cloud gateway. The container is started via the PCS after an experiment is executed, then the data is stored and extracted. If no connection can be established, the data is kept in a queue folder until the connection is restored. Then, the data is uploaded. To know, which data are already uploaded and which are still in the queue, a local data archive of the current products is maintained.

The connection and authentication between the data lake of Azure and the cloud gateway is token-based and takes place exclusively via HTTPS. Therefore, the Docker container is only allowed to communicate via port 443.

The cloud gateway also manages the configuration of the compute clusters in Azure to execute the configured ML job. The communication between the compute cluster and the cloud gateway takes place in the same way. In addition, a device registration, which must be carried out once by a user, is also required for the communication to the compute cluster. This covers IT security aspects and protects the cloud account against unauthorized access.

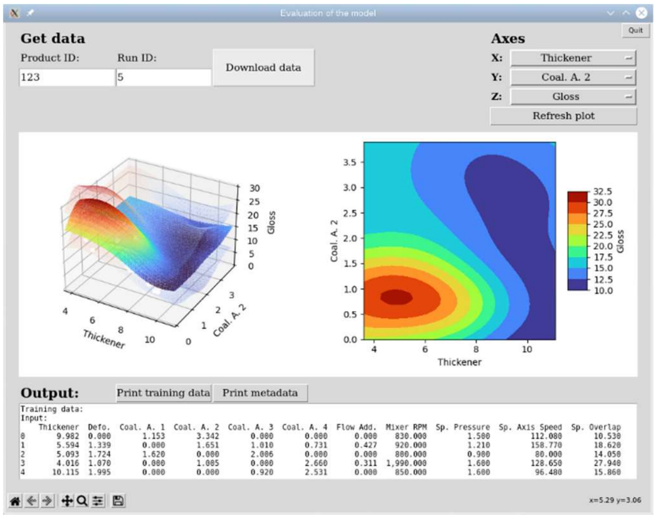
C. Compute Cluster

In this work, compute clusters are registered with Microsoft Azure and used to support data-driven modeling outside of the production environment. In this work, the GP model with ARD is developed using the GPyTorch library [24]. For PyTorch [25], prebuilt docker images already exist on Azure, so there is no need to create an additional image for data-driven modeling.

When a compute cluster is started on Azure, a virtual machine is started. The virtual machine configures a docker container with the required Python packages installed. Then, the job receives the inputs, which contain the training script, the DoE file, and the data of the runs, which are carried out so far. The data of the runs are concatenated to one learning dataset, which is used by the ML algorithm. For scaling the data, the Scikit-Learn library [26] is used. With the knowledge of the DoE file, the constant and dependent parameters are sorted out before learning. The model is then trained until the MLE no longer changes and thus the optimization of the hyperparameters is completed. For the validation of the model, a cross-validation is performed on the dataset. Afterwards the model and the metadata are transferred to the data lake as an output of the job and the compute cluster is closed.

D. Model Deployment and Visualization

The model deployment and visualization of the results also runs in a docker container, like the cloud gateway. Here, the data of a chosen product are downloaded from the data lake via HTTPS and a token based authentication. In Fig. 4, the visualization of the given product feedback is shown. The user interface allows to select a product and a run to download the data.



Product ID: 123
Run ID: 5

X: Thickener
Y: Coal. A. 2
Z: Gloss

```
Training data:
Input:
   Thickener  Defo.  Coal. A. 1  Coal. A. 2  Coal. A. 3  Coal. A. 4  Flow Add.  Mixer RPM  Sp. Pressure  Sp. Axis Speed  Sp. Overlap
0      9.982  0.000       1.153       3.342       0.000       0.000      0.000    830.000         1.500         112.080       10.530
1      5.594  1.339       0.000       1.651       1.010       0.731      0.427    920.000         1.210         158.770       18.620
2      5.093  1.724       1.620       0.000       2.006       0.000      0.000    800.000         0.900          80.000       14.050
3      4.016  1.070       0.000       1.005       0.000       2.660      0.311  1,990.000         1.600         128.650       27.940
4     10.115  1.995       0.000       0.000       0.920       2.531      0.000    850.000         1.600          96.480       15.860
```

Fig. 4: Visualized product feedback

This includes all experimental data up to the selected run and the associated model. The process can be displayed depending on two input parameters and a characterization variable. On the x- and y-axis, input parameters are selected. In this case, these two parameters contain a thickener and a coalescing agent, which are two chemical substances used in the production of coatings. On the z-axis, the corresponding values of the characterization variable are displayed. Here, the gloss of the coating is chosen.

In the left figure, the confidence intervals of the Gaussian process are shown in addition to the predicted values. These indicate in which range the value can lie, i.e., which standard deviation exists in the model prediction. The right figure gives an overview of the height profile of the model mean values. Thus, it is quickly possible to visually determine the minima and maxima of the characterization variables. It is also possible for the user to view the training data and metadata. With this information, it is easier for the chemical engineer to prepare new experiments.

V. Results And Discussion

The presented automation framework allows a flexible and scalable way to integrate data-driven modeling in an industrial environment. The flexibility is given by the fact, that any type of DoE can be handled according to the file structure presented in this paper and independent of the number of input parameters and experiments. In addition, the framework allows the replacement of the modeling algorithm in the case of regression tasks by adapting or replacing the training script in the cloud. The use of a data lake storage as a cloud service offers the possibility of accessing data outside the production environment. This adds further flexibility to the handling of the data. The flexibility and generality of the framework was tested in the context of application in high throughput coating formulation with data from the workference. Therefore, different combinations of input parameters and characterization variables were chosen and the automated modeling process was carried out successfully.

The scalability of this framework is given, as the computing power can be adjusted in the cloud, for example if deep learning is to be used. It is also possible to perform data processing or data-driven modeling on several computing units in parallel and at the same time. Through the usage of a data lake, the framework is not limited to a specific type of data. This means, that the framework is designed in such a way, that other data, like images, can also be integrated for machine learning in the future.

To ensure, that the framework can be used in an industrial environment, the architecture of this framework is based on NOA. With the use of token-based authentication and device registration at Microsoft Azure, a secure way to connect the production environment to the cloud was offered. By using containers, this framework demonstrates the possibility of integrating the latest libraries to interact with innovative cloud technologies in robust production environments. Another advantage of using containers is, that they can be stopped by the PCS if too much computation is required. This ensures the high availability of the CPC domain.

With the visual feedback of the product through the model predictions, it is possible for the chemical engineer to get a sense of the product to be developed. With the Gaussian processes, the chemical engineer is also given feedback on confidence intervals, to know how reliable the model predictions are in the specific regions.

VI. Summary And Future Work

In this work, a framework for automated data-driven modeling in an industrial environment with cloud services is developed. Therefore, the standardized architecture NOA is introduced as an orientation. With this framework, the manual processes of data extraction and preparation from a production environment are automated. Furthermore, the data-driven modeling and the storage of important metadata is integrated in the framework. For modeling, Gaussian processes are used in this work and a visual feedback of the modeling process is given.

In future work, it is possible to extend the framework to different data sources and also to other modeling tasks, like image recognition. Future work should also integrate the DoE into the workflow. Furthermore, the chemical engineer should be assisted in the DoE by the ML algorithm proposing new experiments.

Acknowledgement

The authors acknowledge gratefully Christian Schmitz[1], Kevin Cremanns[2], Carmen Stoffelen-Janßen[1], Lasse Wagner[3], and Gaoyuan Zhang[4] for the data which were generated at the "Coatings Go Digital" workference at the HIT of the University of Applied Sciences Niederrhein (https://smartpaintfactory.com/workference/).

References

[1] K. Bock, “Chemistry 4.0: Growth through innovation in a transforming world,” *Verband der Chemischen Industrie e.V.*, 2017.

[2] B. Strehmel, C. Schmitz, K. Cremanns, and J. Göttert, “Photochemistry with Cyanines in the Near Infrared: A Step to Chemistry 4.0 Technologies,” *Chemistry*, vol. 25, no. 56, pp. 12855–12864, 2019, doi: 10.1002/chem.201901746.

[3] M. Gärtler, V. Khaydarov, B. Klöpper, and L. Urbas, “The Machine Learning Life Cycle in Chemical Operations – Status and Open Challenges,” *Chemie Ingenieur Technik*, vol. 93, no. 12, pp. 2063–2080, 2021.

[4] R. Hai, C. Quix, and M. Jarke, “Data lake concept and systems: a survey,” *http://arxiv.org/pdf/2106.09592v1*, 2021.

[1] Postdoctoral researcher at University of Applied Sciences Niederrhein

[2] Co-founder & chief research and development officer at PI Probaligence GmbH

[3] Research assistant at University of Applied Sciences Niederrhein

[4] Research assistant and doctoral student at University of Applied Sciences Niederrhein

[5] C. Quix, R. Hai, and I. Vatov, "Metadata Extraction and Management in Data LakesWith GEMMS," *CSIMQ*, no. 9, pp. 67–83, 2016.

[6] C. Schmitz, K. Cremanns, and G. Bissadi, "Application of machine learning algorithms for use in material chemistry," in *Computational and Data-Driven Chemistry Using Artificial Intelligence*: Elsevier, 2022, pp. 161–192.

[7] K. Cremanns, C. Schmitz, and L. Wagner, "A computer learns colours - Colouristic based on artificial intelligence," *European Coatings Journal*, pp. 34–39, 2020.

[8] T. Kern, E. Krhač Andrašec, B. Urh, and M. Senegačnik, "Digital Transformation Reduces Costs of the Paints and Coatings Development Process," *Coatings*, vol. 10, no. 7, 2020.

[9] P. Kaushik, A. M. Rao, D. P. Singh, S. Vashisht, and S. Gupta, "Cloud Computing and Comparison based on Service and Performance between Amazon AWS, Microsoft Azure, and Google Cloud," in *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, 2021, pp. 268–273.

[10] NE 175, *NAMUR Open Architecture -- NOA Concept*. Berlin: Beuth Verlag, 2020. Accessed: Jul. 6 2022.

[11] NE 176, *NAMUR Open Architecture -- NOA Information Model*. Berlin: Beuth Verlag, 2021. Accessed: Jul. 6 2022.

[12] NE 177, *NAMUR Open Architecture -- NOA Security Zones and Security Gateway*. Berlin: Beuth Verlag, 2021. Accessed: Jul. 6 2022.

[13] L. Rahm, J. Lorenz, V. Khaydarov, T. Maywald, T. Glas, and L. Urbas, "Predictive maintenance with NOA: Application and insights for rotating equipment," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vienna, Austria, 2020, pp. 761–767.

[14] I. Saeed, S. Baras, and H. Hajjdiab, "Security and Privacy of AWS S3 and Azure Blob Storage Services," in *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, 2019, pp. 388–394.

[15] C. Kotas, T. Naughton, and N. Imam, "A comparison of Amazon Web Services and Microsoft Azure cloud platforms for high performance computing," in *2018 IEEE International Conference on Consumer Electronics (ICCE)*, 2018, pp. 1–4.

[16] M. A. Kamal*, H. W. Raza, M. M. Alam, and M. M. Su'ud*, "Highlight the Features of AWS, GCP and Microsoft Azure that Have an Impact when Choosing a Cloud Service Provider," *IJRTE*, vol. 8, no. 5, pp. 4124–4132, 2020.

[17] C. E. Rasmussen and C. K. I. Williams, *Gaussian process for machine learning*. London, England: The MIT Press, 2006.

[18] J. Keogh and A. Mueen, "Curse of Dimensionality," *Encyclopedia of Machine Learning*, pp. 257–258, 2011.

[19] T. N. Varunram, M. B. Shivaprasad, K. H. Aishwarya, A. Balraj, S. V. Savish, and S. Ullas, "Analysis of Different Dimensionality Reduction Techniques and Machine Learning Algorithms for an Intrusion Detection System," in *2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA)*, 2021, pp. 237–242.

[20] C. Paciorek and M. Schervish, "Nonstationary Covariance Functions for Gaussian Process Regression," in *Advances in Neural Information Processing Systems*, 2003.

[21] C. Williams and C. Rasmussen, "Gaussian Processes for Regression," in *Advances in Neural Information Processing Systems*, 1995.

[22] M. L. Stein, *Interpolation of spatial data: Some theory for kriging*: Springer, 2013.

[23] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference for Learning Representations*, 2015.

[24] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, *GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration.*

[25] A. Paszke *et al.,* "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *33rd Conference on Neural Information Processing Systems 2019.*

[26] F. Pedregosa *et al.,* "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 2011.