Machine Learning for Soft Sensors and an Application in Cement Production

Marcel Stöhr and Thomas Zielke

Abstract Soft Sensors are predictors for measurements that are difficult or impossible to obtain by means of a physical sensor. A soft sensor delivers virtual measurements based on several or many physical measurements and a mathematical or numerical model that incorporates the physical knowledge about the interdependencies. In industrial processes, control, optimization, monitoring, and maintenance can benefit from the application of soft sensors. In recent years, Machine Learning (ML) has proven to be very effective for building soft sensors for industrial applications. We work on modelling dynamical processes for which the measurements of the physical variables are available as time series over long periods. In this article we present work on the comparison of different ML methods for modelling dynamic processes with the objective of predicting certain output variables. The main application focus of our work is on the optimization of the cement production process. We compare approaches using Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) using real process data from a cement production. The cement sector is the third-largest industrial energy consumer. There is a high potential for energy savings and for the reduction of CO_2 emissions by optimizations of the production process. We present first results of our research that primarily is aimed to improve the robustness of a soft sensor for grain size in the cement production process.

Keywords Soft sensors · Machine learning · Dynamic processes · Cement production · Robustness · Artificial intelligence

M. Stöhr (✉) · T. Zielke
Düsseldorf University of Applied Sciences, Düsseldorf, Germany
e-mail: marcel.stoehr@hs-duesseldorf.de

T. Zielke
e-mail: thomas.zielke@hs-duesseldorf.de

K. von Leipzig et al. (eds.), *Smart, Sustainable Manufacturing in an Ever-Changing World*, Lecture Notes in Production Engineering,
https://doi.org/10.1007/978-3-031-15602-1_46

1 Introduction

The objective of a soft sensor is the estimation of a quantity, which cannot or not easily be measured directly. This description may also match a simple resistance thermometer. However, the term soft sensor is only used for inferential measurements that are based on several or many physical measurements and a mathematical or numerical model that incorporates the physical knowledge about the interdependencies. In the literature, the first industrial applications of soft sensors are described in the context of chemical process operations. Rao et al. [1] state two objectives for developing soft sensors: (1) *providing near optimal values for important non-measurable control variables associated with product quality to improve real-time control systems; (2) providing the interpretation of the important process variables for operators to enhance the interaction between chemical processes and human operators* (…).

The objectives of our work are very similar. The processes we deal with are dynamical and the physical measurements are represented by time series of continuous values. We use methods of machine learning to achieve the objectives.

Data-driven approaches for the development of soft sensors have been used for more than twenty years [2]. Early applications of neural networks for this purpose are described in [3–6]. Deep Neural Networks (DNN) allow more complex models that potentially can lead to an improvement of the prediction accuracy [7, 8]. Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) are specific types of DNN. They are particularly apt to capture the information on the dynamics [9, 10] of the available measurements. A recent review on soft sensors in industrial processes [11] provides an excellent overview of the field, far beyond the scope of the brief introduction given here. In modern factories all operational events and all measurements are digitally recorded. This is also the case for the cement mill that we aim to optimize. Figure 1 is an illustration of the cement production process. The mill is filled with fresh and coarse material. After grinding, the material is split by an air separator into new coarse material and the finished product. Since direct measurements inside a cement ball mill are hardly possible, the grain size of the material, a crucial parameter, has to be determined offline. This is a manual operation with a relatively low sampling rate typically about two hours [12]. Based on the measured distribution of the current grain size, a machine operator can adjust the air separator. Ball mills have a high specific grinding energy demand [13]. The reduction of grinding iterations by means of a real-time estimation of the grain size can significantly lower the power consumption.

Previous research has shown that the grain size can be estimated using soft sensors developed with a data-driven approach [12, 14]. A related example is the prediction of the content of free lime (f-CaO) in cement clinker [15]. In practice, a sustainable deployment of this technology has not been achieved yet. The main problems are long-term drifts of process parameters, insufficient robustness with respect to situations not covered by the training data, and lack of transparency of the model behaviour

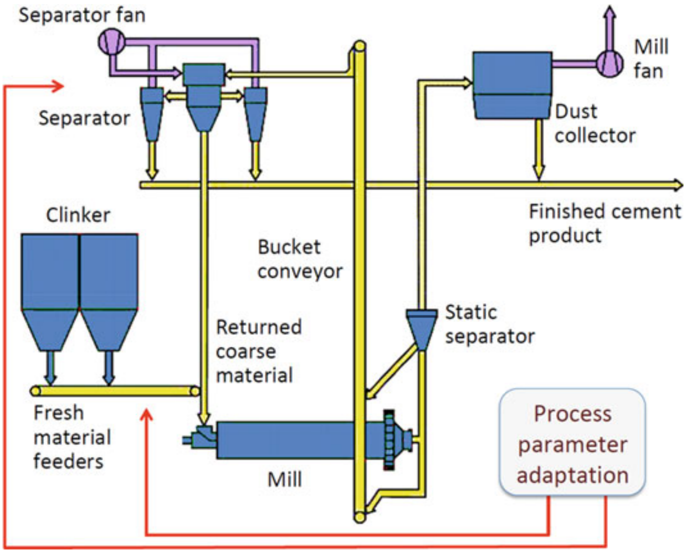


Fig. 1 Cement production process flowchart. (Adapted from en.wikipedia.org/wiki/Cement_mill)

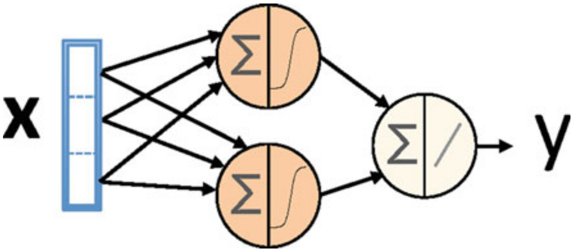
for the responsible process operators. These problems are typical for many applications of artificial intelligence in industry [16]. In Sect. 2 we briefly introduce the RNN and the CNN architectures. Our comparative experiments, described in detail in Sect. 4, use current and historic data from a cement mill and from the operation of a gas-fired absorption heat pump [7].

2 Neural Networks for Time Series Prediction

2.1 Deep Neural Networks (DNN)

Artificial neural networks (ANN/NN) are an established methodology for modeling complex input–output relationships. Figure 2 shows a feedforward neural network, a type of network architecture that is widely used for regression or classification problems. The example network has a three-dimensional input vector **x**, one output value y, and one hidden layer with two neurons. Using a machine learning algorithm, e.g., backpropagation, the network can be trained to produce certain values for y, depending on the given input vectors **x**. The training process can be described as a data-driven optimization of the free network parameters, typically the values for the weights and biases. The network architecture and its trained parameters of the network, data-driven optimized values for the weights and biases, represent a numerical model of the relationship between the inputs and the outputs. A feedforward neural network may have many layers and a large number of free parameters.

Fig. 2 Example of a feedforward network



Such networks are called deep neural networks (DNN). DNN models have a higher expressive power than models built with small networks. Lippel et al. [7] used a DNN for the prediction of output temperatures of a heat pump.

The input measurements are time series data. As shown in [7], the prediction accuracy can be greatly improved when the input vector of the network is augmented by some aggregation of measurements taken at preceding points in time. There are neural network architectures that are especially suitable for the aggregation of information over time. Two of them, recurrent neural networks and convolutional neural networks are briefly described in the following paragraphs.

2.2 Recurrent Neural Networks (RNN)

The typical characteristics of the RNN architecture are feedback loops, at least one. This gives the RNN the capacity to update the current state based on past states and current input data [17]. That is, unlike a feedforward NN the output of a RNN depends on the current input and an internal state, which can also be called a memory. Practical implementations of the feedback loop are based on 'unfolding'. As is illustrated in Fig. 3 only a certain number of past points in time are used by the 'memory'. In our example, only the directly preceding measurement (*t-1*) is used together with the current measurement at time *t*. For our experiments we use a special type of RNN, a so-called 'long short-term memory' network (LSTM). LSTM networks have been widely and successfully used in various applications. An explanation of LSTM is beyond the scope of this article. We refer the interested reader to a review paper by Yu et al. [17].

2.3 Convolutional Neural Networks (CNN)

A convolutional neural network is a special type of feedforward network. CNNs have proven extremely successful for image analysis. The CNN architecture is biologically inspired, since it is known that the visual system is based on 'receptive fields' that play a similar role as filter banks in classical computer vision systems [18]. Recently

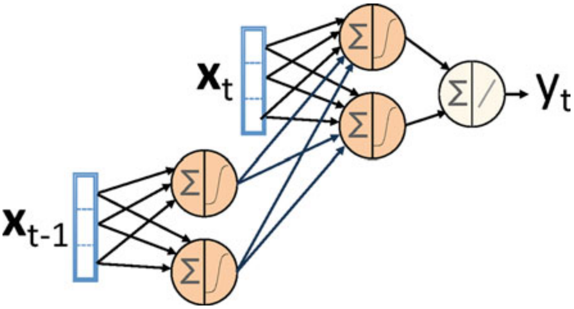


Fig. 3 A simple unfolded RNN with one hidden layer. x is a three-dimensional input vector and y is the output value at time t

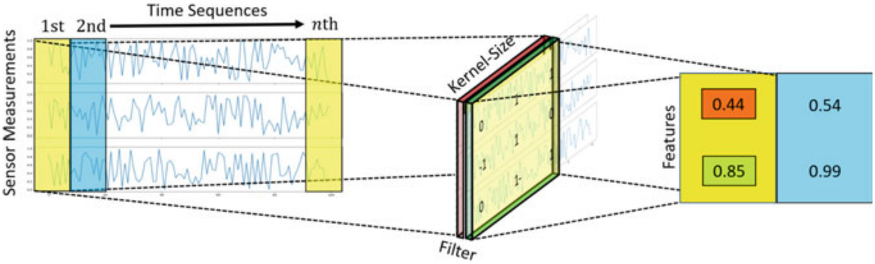


Fig. 4 First convolutional layer of a trained CNN for three 1D time sequences. The kernel slides over all time sequences. For every filter the dot product of the input and the kernel weights is calculated. After the first calculation (yellow) the kernel slides a fixed step size forward and calculates the next value (blue)

this network architecture has been widely used to process 1D signals from production processes [12, 15]. A CNN typically consists of four types of layers: convolutional, pooling, flatten and fully connected layer. The convolutional layer calculates for each filter a dot product of the input vector and kernel weights. This is followed by a so-called pooling layer that reduces and aggregates the raw data. In case of an 'average pooling layer' the amount of data is reduced by averaging the respective inputs. A 'max pooling layer' reduces the amount of data by selecting the largest value. After each calculation this filter window slides forward across the layer inputs. The fully connected layer at the end of the CNN, will predict the target value (Fig. 4).

3 Auxiliary Methods

3.1 Metrics

The most common used error metric is the mean squared error (MSE), which weights outlier stronger. The mean absolute percentage error (MAPE) is also a good metric to compare models on different datasets. Using MAPE has the advantage that the absolute values of the underlying data is not made explicit. In this article the mean absolute percentage error is used to compare results between models and datasets [19]:

$$MAPE = \frac{\sum \frac{\left|y_{true} - y_{pred}\right|}{y_{true}}}{n} \cdot 100 \tag{1}$$

where y_{true} denotes the ground truth, y_{pred} the prediction and n the number of predictions. There are a number of other metrics that could also be considered as an alternative to MSE [20].

3.2 Autoencoder for Anomaly Detection

An often-overlooked problem are possible anomalies and outliers in the data. In the following two methods for the detection of anomalies and outliers are described. The so called Autoencoder is special neural network architecture that consists of two parts. The first part encodes the input sequence by reducing dimensions, the second is a decoder that ideally reproduces the input data. Using the training data as reference, outliers or anomalies are detected by a reconstruction of the test data and calculating the deviations between input and output data of the Autoencoder [21].

4 Experiments

The comparison of the different neural network architectures is made with real process data from a cement production plant and from the operation of a gas-fired absorption heat pump [7]. Hyperparameter optimization for the number of filters (CNN) and cells (LSTM) is performed for both models and datasets. For the heat pump the objective is to predict the outlet temperatures for heating T_h and cold-water T_c circuit based on five input variables such as the volume flow rate of used gas, inlet temperature and volume flow rate of the heating and cold-water circuit. In the cement production process the task is to predict two parameters of the Rosin–Rammler-Sperling-Bennett RRSB distribution [22, 23]. In the following sections the results for the hyperparameter search and the reaction of the models on outliers are shown. For this article a CNN like LeNet-5 is used which consists of two convolutional layers followed by a pooling layer and three fully connected layer with a size of 120, 64 and 2 [24]. In both experiments the prediction models are the same and are built according to Table 1. The datasets differ in the number of training data, validation data, and number of features, while the number of targets is two for both models. The heat pump dataset consists of 1.2 million training data points and 950 thousand validation data points each with five input parameters. In contrast, the

Table 1 Concrete model structures for CNN and LSTM used in the experiments. The fully connected layers form the end for both models. F_{convA} and F_{convB} denote the number of filters and $Cell_A$ and $Cell_B$ describe the number of LSTM cells

CNN		*LSTM*	
Layer	Output	Layer	Output
Convolutional	(1200, F_{convA})	LSTM	(1200, , $Cell_A$)
Avg. pooling	(600, F_{convA})	LSTM	($Cell_B$)
Convolutional	(600, F_{convB})	–	–
Avg. pooling	(300, F_{convB})	–	–
Flatten layer	($300 \cdot F_{convB}$)	–	–
Both			
Layer		Output	
Fully connected		(120)	
Fully connected		(84)	
Fully connected		(2)	

Table 2 The best achieved MAPE of the hyperparameter study. The numbers in parenthesis describe the used parameters and random seed

Model	Heat pump	Cement
LSTM	0.77% (32;16; $Seed_2$)	2.14% (64;4; $Seed_1$)
CNN	1.15% (64;16; $Seed_1$)	2.27% (16;32; $Seed_3$)

cement dataset consists of only 5500 and 605 target data points but with 19 input parameters.

To train the weights in neural networks, initial values must be defined at the beginning, which are usually set ‘randomly’. The random seed defines the random status to produce reproducible results. In addition, different seeds can be tried out to identify a particularly bad or good initialization (Table 2).

4.1 Hyperparameter Grid Search

In this section the results of the parameter optimization of both datasets will be presented. The selection of the hyperparameters to be optimized remains the same for both datasets, while only the random seed changes, since the random selection of the weight initialization can have a strong influence on the result. The hyperparameters for these experiments are shown in the Table 3.

Initial results with the heat pump dataset in Fig. 4 showed that the CNN (orange) mostly produced better and more robust results. However, it also shows that different random seeds are important. The best MAPE according to Table 3 from the LSTM in the first seed is 2.23% and in the second 0.77%. With the cement dataset the LSTM

Table 3 Hyperparameter for the grid search. *Grid A* are the different sizes for the first layer and *Grid B* for the second

	CNN/LSTM			
Grid A	8	16	32	64
Grid B	8	16	32	64

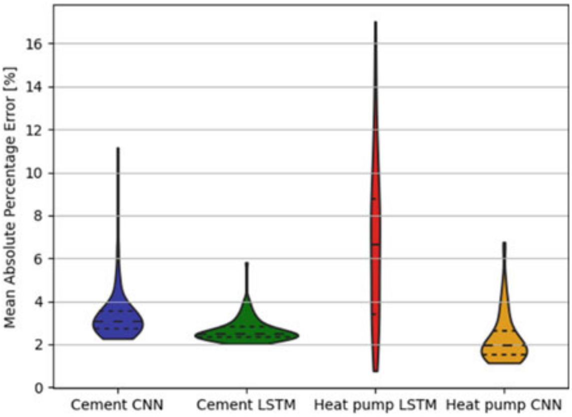


Fig. 5 Violin plot from MAPE of both models and datasets over all seeds

seems to be more stable but with a worse best MAPE then the CNN. Figure 5 shows a violin plot for both network architectures and both data branches. It visualizes the results summarized for the 3 random seeds.

The peaks of the violins denote the highest and lowest MAPE. The width indicates the distribution of the values and the 3 lines the 3 quartiles. This diagram suggests that in the case of the cement dataset, the number of LSTM cells and the selection of the random seed does not have a large influence on the result. The distribution for the CNN-based model looks similar on both datasets, only shifted in height. While the results from the LSTM for the heat pump scatter strongly regardless of the parameter choice, the cement dataset shows more consistent and better results than the CNN. The Table 3 provides two interesting insights: The best MAPE is very similar for the cement dataset for both architectures, while for the heat pump dataset the LSTM gives a much better result (Fig. 6).

4.2 Reaction to Outlier

The other series of experiments investigate the behaviour of the models to outliers in the data. For this, we manipulated the dataset and added artificial outliers by replacing 50,000 contiguous data points (12% of input data) from 5 features with the max value of the respective feature. Before the forecast, an attempt is made to detect the outliers

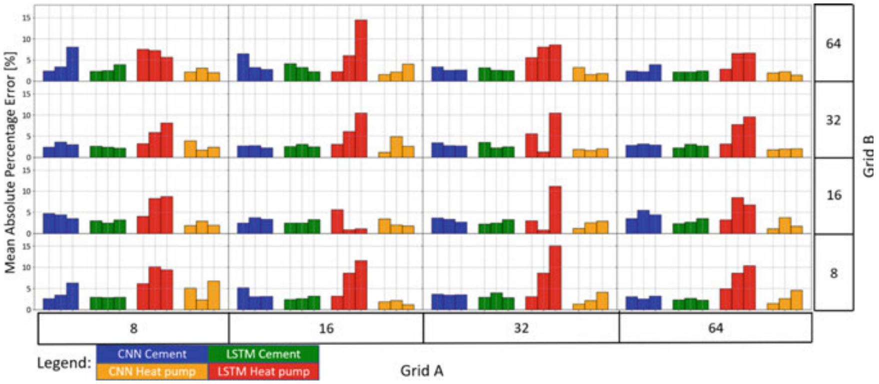


Fig. 6 Results of the hyperparameter study for both networks (CNN/RNN) and datasets (cement/heat pump) in MAPE for 3 random seeds

and replace them with the median of the last 3 days. For this purpose, an autoencoder (AE) is used. The Fig. 7 shows the deviation from the ground truth for the RRSB_D prediction, where the grey area indicates data manipulation. It is noticeable that the LSTM shows significantly better results, especially in the manipulated area beside one outlier.

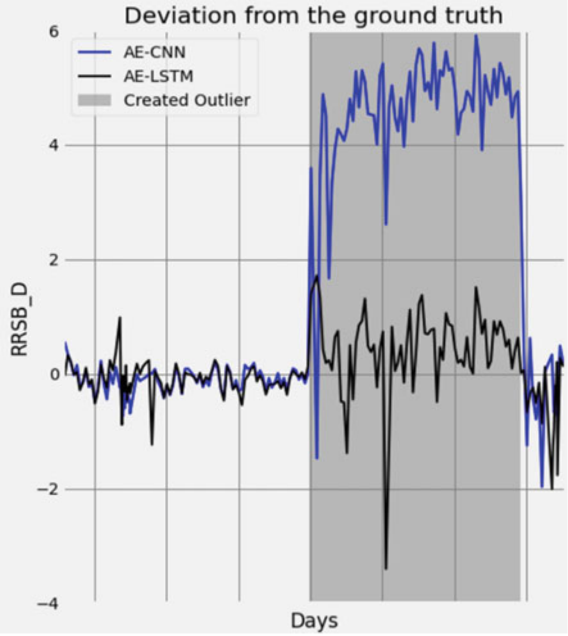


Fig. 7 Reaction of the methods to outlier. Two curves showing the deviation from the ground truth for the target value while a data manipulation is taking place (dark grey area)

5 Conclusion

Machine learning methods have proven to be very effective for building soft sensors. In this article, we study two neural net architectures that are specifically suitable for modelling time series data. Both CNN and RNN are used with real sensory data from two different application domains. The one dataset was collected from a cement production process. The other dataset came from the operation of a gas-fired heat pump.

The trained models were assessed by the mean absolute percentage error (MAPE) between the predicted values and the ground truth data. We achieved high accuracies for both CNN and RNN models. The training of the models was conducted many times using different hyperparameters and various numerical training initialisations. The respective accuracies of the trained models is not the only relevant criterium. We also looked at the robustness of the models in the presence of outliers and anomalies, i.e., how good are the predictions when the time sequence contains abnormal values. The results on the cement dataset can be considered more robust with the CNN models and the RNN models show better robustness on the heat pump dataset. In practical applications, a soft sensor should be operating in concert with a detector for outliers and anomalies. Beside the comparison of CNN and RNN, we presented some preliminary work on the integration of such detectors. For the goal of robust and sustainable applications of soft sensors in complex industrial processes we see a large potential for a fusion of different machines learning methods that may operate in an ensemble or supportively act as decision aids for handling special situations.

Acknowledgements The authors acknowledge the financial support by the German Federal Ministry for Economic Affairs and Energy (BMWi) in the framework of Industrial Collective Research (IGF), Project 21152N/2. We are thankful to Philipp Fleiger who took the initiative for this project. We would like to thank Jens Lippel, Carsten Gallus, and Martin Becker for valuable discussions and their suggestions for the improvement of this article.

References

1. Rao, M., Corbin, J., Wang, Q.: Soft sensors for quality prediction in batch chemical pulping processes. In: Proceedings of 8th IEEE International Symposium on Intelligent Control, pp. 150–155. IEEE (1993)
2. Kadlec, P., Gabrys, B., Strandt, S.: Data-driven Soft Sensors in the process industry. Comput. Chem. Eng. **33**(4), 795–814 (2009)
3. Sevilla, J., Pulido, C.: Virtual industrial sensors trough neural networks. Demonstration examples in nuclear power plants. In: IMTC/98 Conference Proceedings. IEEE Instrumentation and Measurement Technology Conference. Where Instrumentation is Going (Cat. No.98CH36222), pp. 293–297. IEEE (1998)
4. Casali, A., Gonzalez, G., Vallebuona, G., Perez, C., Vargas, R.: Grindability soft-sensors based on lithological composition and on-line measurements. Miner. Eng. **14**(7), 689–700 (2001)

5. Chella, A., Ciarlini, P., Maniscalco, U.: Neural networks as soft sensors: a comparison in a real world application. In: The 2006 IEEE International Joint Conference on Neural Network Proceedings, pp. 2662–2668. IEEE (2006)
6. Luo, J.X., Shao, H.H.: Developing soft sensors using hybrid soft computing methodology: a neurofuzzy system based on rough set theory and genetic algorithms. Soft Comput. **10**(1), 54–60 (2006)
7. Lippel, J., Becker, M., Zielke, T.: Modeling dynamic processes with deep neural networks: a case study with a gas-fired absorption heat pump. In Proceedings of 9th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, pp. 317–326. SCITEPRESS (2019)
8. Maschler, B., Ganssloser, S., Hablizel, A., Weyrich, M.: Deep learning based soft sensors for industrial machinery. Procedia CIRP **99**, 662–667 (2021)
9. Ke, W., Huang, D., Yang, F., Jiang, Y.: Soft sensor development and applications based on LSTM in deep neural networks. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–6. IEEE (2017)
10. Jian, H., Lihui, C., Yongfang, X.: Design of soft sensor for industrial antimony flotation based on deep CNN. In: 2020 Chinese Control and Decision Conference (CCDC), pp. 2492–2496. IEEE (2020)
11. Jiang, Y., Yin, S., Dong, J., Kaynak, O.: A review on soft sensors for monitoring, control, and optimization of industrial processes. IEEE Sensors J. **21**(11), 12868–12881 (2021)
12. Schweikardt, F., Spenner, L.: Maschinelles Lernen: Entwicklung eines Softsensors zur Vorhersage der Mahlfeinheit einer Zementmühle, Karlsruhe Institute of Technology (KIT) (2017)
13. Schneider, M., Romer, M., Tschudin, M., Bolio, H.: Sustainable cement production—present and future. Cem. Concr. Res. **41**(7), 642–650 (2011)
14. Andreatta, K., Apóstolo, F., Nunes, R.: Soft sensor for online cement fineness predicting in ball mills. In: Proceedings of the International Seminar of Science and Applied Technology (ISSAT 2020). Atlantis Press (2020)
15. Jiang, X., Yao, L., Huang, G., Qian, J., Shen, B., Xu, L., Ge, Z.: A spatial-information-based semi-supervised soft sensor for f-CaO content prediction in cement industry. In: 2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS), pp. 898–905. IEEE (2020)
16. Zielke, T.: Is artificial intelligence ready for standardization? In: Yilmaz, M., Niemann, J., Clarke, P., Messnarz, R. (eds.) Systems, Software and Services Process Improvement. Communications in Computer and Information Science, pp. 259–274. Springer International Publishing, Cham (2020)
17. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. **31**(7), 1235–1270 (2019)
18. LeCun, Y., Kavukcuoglu, K., Farabet, C.: Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems, pp. 253–256. IEEE (2010)
19. Khair, U., Fahmi, H., Hakim, S.A., Rahim, R.: Forecasting error calculation with mean absolute deviation and mean absolute percentage error. J. Phys.: Conf. Ser. **930**, 12002 (2017)
20. Fortuna, L., Graziani, S., Xibilia, M.G.: Comparison of soft-sensor design methods for industrial plants using small data sets. IEEE Trans. Instrum. Meas. **58**(8), 2444–2451 (2009)
21. Zhou, C., Paffenroth, R.C.: Anomaly Detection with Robust Deep Autoencoders. In: Proceedings of 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 665–674. ACM (2017)
22. DIN German Institute for Standardization: 1976-04-00. Graphical representation of particle size distributions; RRSB-grid 19.120. Beuth Verlag GmbH, Berlin 19.120, DIN 66145
23. Gao, P., Zhang, T.S., Wei, J.X., Yu, Q.J.: Evaluation of RRSB distribution and lognormal distribution for describing the particle size distribution of graded cementitious materials. Powder Technol. **331**, 137–145 (2018)
24. Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D.J.: 1D convolutional neural networks and applications: a survey. Mech. Syst. Signal Process. **151**, 107398 (2021)

Marcel Stöhr holds a Bachelor of Engineering degree in mechanical engineering and is currently working as a research assistant in the development of robust artificial intelligence at the university of applied sciences in Düsseldorf.

Thomas Zielke holds degrees in computer engineering, knowledge based systems and neural computation. He is a professor of computer science at the Düsseldorf University of Applied Sciences, Germany. His current research interests include pattern recognition, machine learning, and artificial intelligence.