Machine Learning Soft Sensor for Predicting the Product Quality of a Hydrogen Reduction Furnace

Tommi Mäkelä[1], Jari Luomala[2], Leevi Lind[2], and Ismo Hakala[2]

Abstract— **Soft sensors can predict hard-to-measure product quality variables in industrial processes using easy-to-measure variables, such as temperatures and flow rates, as their inputs. Recently, a popular trend in constructing soft sensors has been the data-driven approach, where the sensor can be a machine learning model trained on real process data. This work relates the construction of a machine learning based soft sensor to predict quality indicators in an industrial process centered on a hydrogen reduction furnace. The input data for the soft sensor consist of various measurements that are used to monitor the furnace during operation. A machine learning model, a deep neural network, is trained to predict two product quality indicator parameters from the input data. The performance of the model is tested with previously unseen samples. The tests show that the model performs well, with mean absolute percentage error (MAPE) values of the predictions staying below 3.7% for one of the indicators and below 1.3% for the other.**

I. Introduction

Modern industrial process control typically relies on a wide variety of measurements [1]. These measurements may be divided into online measurements that provide real time data from sensors integrated with production equipment, and into laboratory analyzes performed on samples of products [2]. Typically, the real time measurements provide information on the conditions within the production equipment in the form of some easy-to-measure quantity, such as temperature or pressure. While it would naturally be advantageous to also make real time measurements of variables directly related to product quality, the lack of suitable sensors or their cost often prohibits this [2]. In cases where online product quality measurements are not feasible, quality indicators can instead be quantified with laboratory measurements. However, such measurements may have a significant delay and a monetary cost associated with each produced data point [2].

In addition to laboratory measurements, product quality indicators may be evaluated by constructing a soft sensor: a piece of software that can predict product quality from the real time measurements associated with it [1], [2]. There are two primary ways to construct soft sensors. The first principles approach attempts to model the process from first principles, relying on in-depth knowledge of its physical and chemical properties. In contrast, the data-driven approach does not model the industrial process directly, but tries to learn the relations between real time measurements and product quality from historical data [3], [4]. Often the model employed as a data-driven soft sensor is based on machine learning. In recent years, the data-driven approach has gained popularity, likely due to its comparative ease of implementation and the growing popularity of machine learning in general. In many industrial plants, the data gathered with numerous sensors has been stored in data banks over several years, often without a clear vision on how to utilize it: such data could be seen as a well of untapped potential for constructing soft sensors.

This work relates a process for constructing a machine learning soft sensor for an industrial application centered on a hydrogen reduction furnace, used in the production of cobalt powders. Cobalt is one of the critical materials defined by the European Critical Raw Materials Act proposed in 2023 [5], and is used, for example, in the production of high performance metal alloys, batteries, permanent magnets, and ceramics. The furnace considered in this study is equipped with various sensors providing real time data for monitoring its operation, but the quality of its products is only assessed through laboratory analyzes twice each day. The primary research question of this study is whether data connected to the furnace could be used to construct a machine learning soft sensor, which would provide a surrogate for the laboratory analyzes. If the prediction results from the soft sensor are accurate enough, they could be used to guide the furnace operation at times when the laboratory analyzes are not available. While the methods are mainly data-driven, the approach also makes use of operator experience when deciding which parameters should be used as inputs of the machine learning model. The model presented in this study reached mean absolute percentage error (MAPE) values below 3.7% for one of two quality indicator parameters and below 1.3% for the other. Based on the results, the industrial plant where the data was gathered is considering implementing multiple similar soft sensors into their process.

II. Development of Machine Learning-Based Soft Sensor

This section describes the data on which the soft sensor is built, the preprocessing steps performed on the data, and the machine learning model and its optimization.

A. Data and preprocessing

The data in this study are related to a specific hydrogen reduction furnace, which is used in the manufacture of

*The research for this paper was financially co-funded by the European Union (grant no. A80436).

[1]Tommi Mäkelä is with Jervois Finland, Kokkola, Finland tommi.makela@jervoisglobal.com

[2]Jari Luomala, Leevi Lind, and Ismo Hakala are with the Kokkola University Consortium Chydenius, University of Jyväskylä, Kokkola, Finland jari.t.luomala@jyu.fi, leevi.j.j.lind@jyu.fi, ismo.hakala@jyu.fi

cobalt powder products. Raw material enters this furnace as batches of powder on plates from one side and travels through the furnace, finally exiting from the other side. Each batch spends a predefined time in the furnace. The furnace is equipped with various sensors used to monitor the operation in real time. These sensors measure, for example, temperatures in multiple points of the furnace. Beside the real time measurements, the furnace operation is monitored by laboratory measurements of three quantities: the apparent density (g/cm^3) of the raw material measured once per day, and the oxygen content (%) and surface area (m^2/g) of the furnace output product measured twice per day. The aim of this study was to create a soft sensor that would predict the product quality indicators of oxygen content and surface area from the apparent density measurement and the real time measurements.

During preliminary examination of the data, the laboratory analysis data for oxygen content and surface area were found to have a considerable amount of noise, likely originating from the way the samples for laboratory analysis were gathered. The furnace consists of sections where the reduction takes place under controlled conditions. Processed material is collected and sampled after the furnace and analyzed in a laboratory. Preliminary tests with noisy data resulted in poor performance by the proposed soft sensor. To mitigate this, it was decided to smooth the laboratory analysis data using an unscented Kalman filter [6]. Parameters of the implemented filter were not optimized for each application, but the same approximate set of parameters were used in all applications of the filter.

The data used in this study originate from a store of data gathered on the furnace operation during several years of use. These data were sampled for a total period of 14 months, during which the furnace was in operation for 259 days. Within this time, the furnace was mainly used to make two products, henceforth referred to as Product 1 and Product 2. It was decided to separate the data according to product, and build two product-specific soft sensor models. It was also recognized by operators of the furnace, that not all of the real time measurements correlate with the product quality indicators. Based on the operators' multiple years of experience in running the furnace, the total number of input parameters was limited to 19 for Product 1, and to 20 for Product 2. The selected parameters included the weights of the batch plates, hydrogen flow rate into the furnace, multiple temperature measurements, time between loading subsequent batches, and the apparent density.

Different sensors and other data sources provide data at different rates. To get compatible data that is synchronized with respect to time, all real time measurement values were fetched from the data bank of the industrial plant as averages over a one minute time span. Measurements of apparent density, oxygen content, and surface area were made much more infrequently than the other quantities: apparent density once per day and oxygen content and surface area twice per day, as illustrated in Fig. 1. The laboratory data were resampled so that there would be laboratory data for each

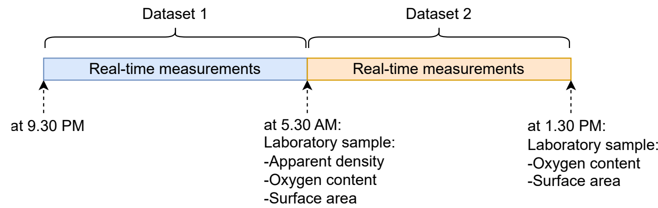

Fig. 1. Formation of datasets with real time data and laboratory analyses.

one minute interval of real time data. First, 8 hours was chosen as an approximate process time — the time that processed material spends in the furnace between loading it in and it passing the point where samples are taken for analysis of oxygen content and surface area. As the simplest possible resampling approach, the values of apparent density, oxygen content, and surface area were copied to obtain corresponding values for each one minute interval. The values of oxygen content and surface area were considered to represent 8 hours of preceding real time measurements, while the values of apparent density were considered to represent 8 hours of real time data before and after the measurement was made, totaling 16 hours.

The resampled data was next examined for missing values and outliers. In the case of missing laboratory values, the real time measurements connected to the missing value were removed from the data. Missing real time measurement values were removed, if the loss of data was caused by the entire furnace being out of commission, for example due to a planned service break. If only some of the sensors providing real time data had missing values, the data was removed only if the sensor was out of commission for longer than 6 hours. For shorter periods with no data, the missing values were replaced with the last recorded value: the sensor readings are assumed to change slowly during normal operation of the furnace, which yields validity to this approach.

Obvious outliers could be detected in the real time data by setting limits to the normal operating conditions of the furnace: for example, the temperature readings are expected to always stay between certain threshold values when the furnace is in operation. Values below the lower limit were replaced with the lower boundary value, and values above the upper limit were replaced with the upper boundary value. Less obvious outliers were detected by setting limits using the interquartile range (IQR) of each variable. The IQR is calculated by first dividing the data into quarters, with 25% of the samples between the minimum value and a lower quartile value $q1$, 25% between $q1$ and the median value, another 25% between the median value and an upper quartile value $q3$, and the final 25% between $q3$ and the maximum value. The IQR is defined as the range between the quartiles $q1$ and $q3$: $\mathrm{IQR} = q3 - q1$. A data point p was flagged as an outlier if the following condition was true:

$$p > q3 + [1.5(\mathrm{IQR})] \mid p < q1 - [1.5(\mathrm{IQR})]. \quad (1)$$

The detected outliers were not removed from the data, but replaced with the boundary value closer to them.

B. Model

The machine learning model employed in this study was a multilayer perceptron (MLP), a common architecture for artificial neural networks [7]. With their large number of adjustable variables, deep neural networks are able to learn complex nonlinear relations. The main drawbacks of these models are the large amounts of data and computing power required in their training. In recent years, the availability of both data and suitable computer hardware has increased, leading to a surge in the popularity of neural networks in many fields. Various neural network architectures have also been used to build data-driven soft sensors for industrial process control [4], [8]–[12].

The neural network was trained in a supervised learning scheme, where each output of the network is compared to a ground truth value. Each input of the network consisted of 19 or 20 values, depending on the product. The input variables included the time between loading subsequent plates into the furnace, a laboratory analysis result of the apparent density of input material and real time measurements on temperatures, hydrogen flow rate, and the weights of the plates. Additionally, the preceding laboratory analysis results of apparent density, oxygen content, and surface area were included in the input: besides the inclusion of these values, the time series nature of the data was not utilized, treating each sample as a separate point. The network was configured to output two values, one for oxygen content and one for surface area: during training, these output values were compared against their corresponding ground truth values, and the network weights updated according to the comparison.

After the preprocessing steps of resampling, dealing with missing values, and removing outliers, the data were partitioned into separate sets for training and testing. For rigorous testing of the presented method, a section of data from a certain time period was reserved only for testing. A convenient method for selecting the testing partition was provided in the method that the furnace is operated with, as it is used to run campaigns of the two products: a few whole campaigns of both products were designated for testing, and the rest were used for training and optimizing the method. These testing sets were dubbed unseen test data. The training sets were further divided into subsets for training, validation, and testing. Here, the validation sets were used to examine the performance of the method during training runs, to determine when the model starts to overfit to the training data. The testing sets were used to examine and compare the network performance with different hyperparameter combinations, in order to optimize the hyperparameters. The unseen testing data was only used for final testing after the networks were optimized. The percentages and sample counts in each subset of data are detailed in Tab. I.

The model of choice in this study was an MLP with three hidden layers. This model was implemented with the Python programming language, using the Keras [13] framework of TensorFlow [14], a popular library for constructing neural networks. The input data was scaled before feeding it to the network. Two scaling approaches from Scikit-learn [15] library, MinMaxScaler and StandardScaler, were tried, with the choice of scaling method being left as an adjustable hyperparameter. For both methods, the scaling parameters were determined from the training data and saved for later use: the scaling of all inputs was performed using these values. The network was set to train for a maximum of 500 epochs, but the training was stopped earlier if the performance with validation data did not improve for 5 subsequent epochs: the final model for Product 1 completed its training in just 14 epochs, while the model for Product 2 took 83 epochs.

Grid search was used to optimize hyperparameters of the models, leading to slightly different results for the two products. In total, 72 models with different hyperparameter values were trained for each product. A summary of the final hyperparameters may be seen in Tab. II. An illustration of the network with its inputs and outputs is presented in Fig. 2.

TABLE I

SPLITTING OF DATA INTO SETS FOR TRAINING AND TESTING, WITH PERCENTAGES OF DATA AND THE COUNTS OF SAMPLES IN EACH SET.

	Product 1		Product 2	
Dataset	Percentage	Samples	Percentage	Samples
Training	45	37 070	61	89 170
Validation	11	9 268	15	22 292
Test	14	11 284	15	22 592
Unseen test	30	24 643	9	12 458

III. PERFORMANCE EVALUATION

The performance of the final models was evaluated with previously unseen data gathered during a different time period than the data used in the training phase. As with the training data, the ground truth values of the unseen test data were filtered using a Kalman filter. The same filtering was also applied to the network outputs to render them comparable with the ground truth. Unlike in the training phase, the laboratory unseen test data was not resampled to the same one minute grid that was used for the real

TABLE II

HYPERPARAMETERS OF THE TWO OPTIMIZED NEURAL NETWORKS. ROWS BELOW THE SINGLE HORIZONTAL LINE RELATE TO HYPERPARAMETERS THAT UNDERWENT OPTIMIZATION WITH GRID SEARCH.

Hyperparameter	Optimized: Product 1	Optimized: Product 2
Optimizer	Adam	Adam
Learning rate	0.001	0.001
Batch size	32	32
Max. epochs	500	500
Activation	ReLU	ReLU
Hidden layer count	3	3
Scheduler	ExponentialDecay	InverseTimeDecay
Loss fn.	Mean Square Error	Mean Absolute Error
Scaling	MinMaxScaler	MinMaxScaler
Dropout	0.1	0
Node counts	[250, 125, 60]	[125, 60, 30]

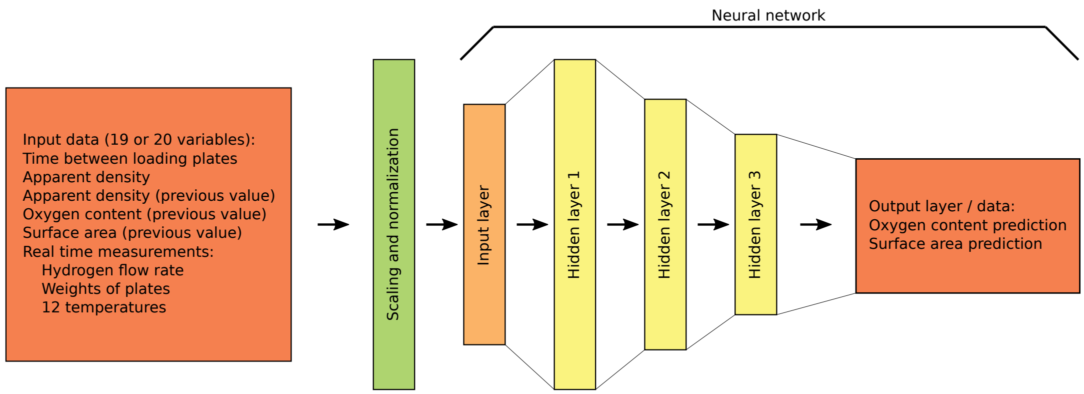

Fig. 2. Illustration of the constructed neural network together with its input and output variables.

time measurements. Instead, the network was used to predict oxygen content and surface area for all 480 one minute intervals that correspond to the 8 hour period designated to each set of laboratory values. The 480 network predictions were then averaged to render them comparable with the corresponding ground truth laboratory value.

The model performance was quantified by calculating the mean absolute percentage errors (MAPE) and mean absolute errors (MAE) of the predictions and by plotting the predictions with respect to time together with the ground truth to qualitatively examine their differences. The MAPE and MAE values of the two quality indicator predictions for the two products are presented in Tab. III and the plots in Fig. 3.

According to the MAPE and MAE values, the predictions for Product 1 were more accurate than the predictions for Product 2. This is interesting, as Product 2 had more samples in the training set than Product 1. While an attemt to avoid overfitting the model to the training data was made by applying early stopping in the training, this attempt may not have been successful. It is also possible that the quality indicators of Product 2 are simply more difficult to predict than those of Product 1. Yet another possible reason can be found in the hyperparameter optimization performed with grid search, where one of the two products may have found more suitable hyperparameter values than the other. Finally, it is also possible that for Product 2 the correlations between the real time measurements and the laboratory measurements are simply weaker than for Product 1, making the quality of Product 2 harder to predict.

The MAPE values of the oxygen content predictions are higher than those of surface area predictions for both products. One reason for this could be in the choice of cost function, which was mean square error for Product 1 and mean absolute error for Product 2. As seen in Fig. 3, the numerical values for oxygen content are lower than those for surface area. The error metrics do not take this into account, thus placing more weight on the surface area prediction. Using a relative metric, such as MAPE, in the cost function could have yielded more balanced results. Another reason could be in delays between gathering a sample and measuring it in the laboratory. The surface area of a sample will typically not change with time, but if the sample is in contact with air, the oxygen content can increase. With irregular delays, this would manifest as noise in the data, making the oxygen content harder to predict.

TABLE III

MEAN ABSOLUTE PERCENTAGE ERROR (MAPE) AND MEAN ABSOLUTE ERROR (MAE) VALUES OF THE NEURAL NETWORK SOFT SENSOR PREDICTIONS FOR THE TWO PREDICTED QUALITY INDICATORS, FOR TWO PRODUCTS.

	Oxygen		Area	
	MAPE	MAE	MAPE	MAE
Product 1	2.137	0.005	1.101	0.014
Product 2	3.638	0.007	1.211	0.010

IV. DISCUSSION

While the presented soft sensor has been shown to be capable of performing well, deploying the model in a production environment comes with its own challenges. For example, there is naturally a delay between gathering the samples for laboratory analysis, and the analysis data being uploaded to the database of the industrial plant. This delay may prove problematic, since the method relies on using the previous analysis results as inputs of the network. Deploying the model in practice gives an interesting line of further research, which will provide insight into the concrete advantages that soft sensors can offer.

One largely open question posed by this study is the generalizability of the developed models. From the point of view of an industrial plant, it could be advantageous to have only one general model that can give predictions for all their products, from all their furnaces. In this preliminary study, a choice was made to focus on a single furnace and implement separate models for two products made using the furnace. Future research could aim toward a more general model that could predict the quality indicators of multiple products.

A multilayer perceptron is hardly the only machine learning model that could be employed as a soft sensor in the case studied in this work. Preliminary results from applying gradient boosting [16] to the same problem have been promising. Compared to the MLP, gradient boosting requires

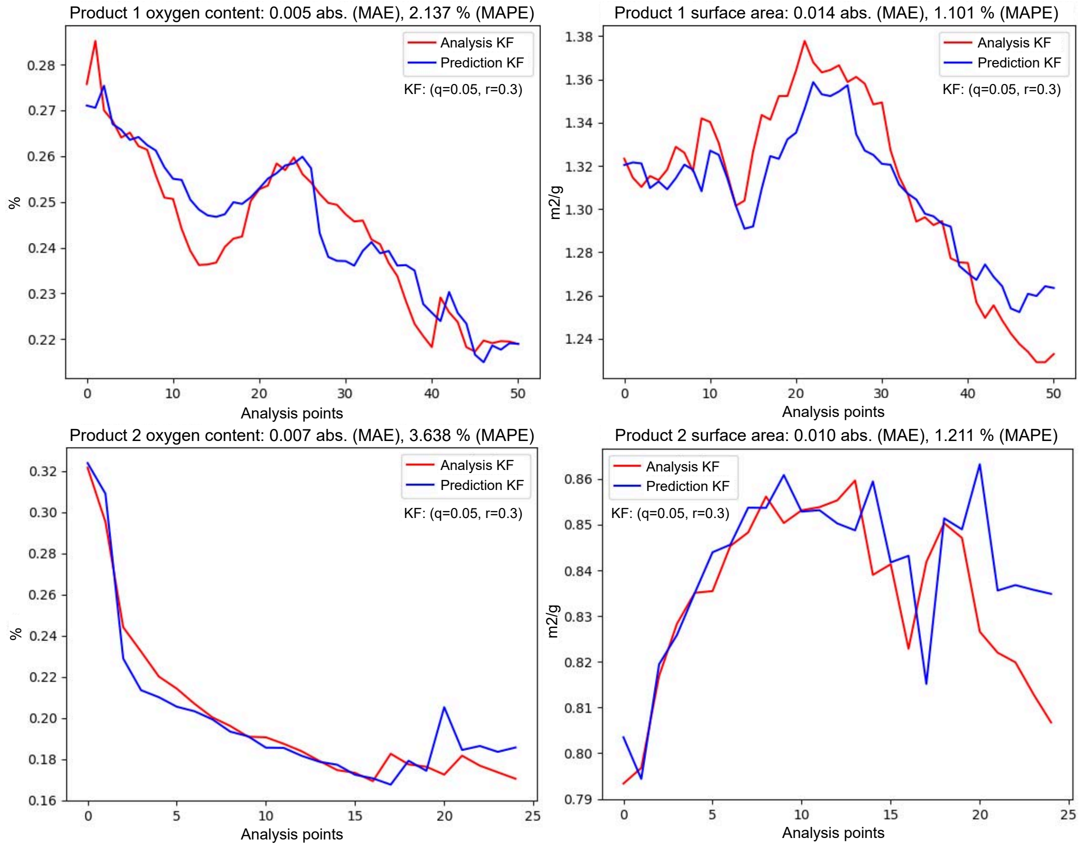

Fig. 3. Soft sensor predictions and corresponding ground truth for the two predicted quality indicators, for two products. The samples are arranged according to their timestamps.

much less computing power, and would thus be a good candidate for a model that is constantly retrained as new data comes in. Other models that could be tried include the long short-term memory (LSTM) neural networks [17], and neural networks based on the gated recurrent unit (GRU) [18]: these both are particularly well suited for time series data, and could yield better results by leveraging the temporal relationships found in industrial process data.

Another consideration for future versions of the developed soft sensor is its explainability. At the moment, the implementation is effectively a "black box" that offers no insight on how it reaches a certain value for its prediction. From the point of view of an operator responsible for controlling the reduction furnace, such predictions may not seem trustworthy. Like laboratory measurements, they also do not necessarily provide guidance on how to adjust the process for a better outcome. One way to correct this would be to calculate a feature importance score for each input variable, to compare the contribution of the different inputs. This would allow the operators to see how the different process variables affect the predicted quality indicator parameters. Certain classes of machine learning models are better suited for explainability than others, and this could also affect the choice of model in future versions. For example, calculating the feature importance for a gradient boosting model built using XGBoost [19] can be done with a single command.

V. Conclusion

This work introduced a neural network based software sensor for a hydrogen reduction furnace, discussing the data gathering for training, the development of the model, and presenting results from a performance evaluation with data previously unseen by the model. Overall, the achieved results may be seen as very encouraging. The mean absolute percentage errors of both predictions listed in Tab. III remained below 4% for data previously unseen by the network. Moreover, the network predictions were largely consistent with the ground truth, also when examined with respect to time, as presented in Fig. 3. Perhaps the most convincing endorsement for the presented method comes from the industrial plant where the data was gathered, as they are considering incorporating multiple similar soft sensors in the production line.

References

[1] P. Kadlec, B. Gabrys, and S. Strandt, "Data-driven soft sensors in the process industry," *Computers & Chemical Engineering*, vol. 33, no. 4, pp. 795–814, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135409000076

[2] F. A. A. Souza, R. Araújo, and J. Mendes, "Review of soft sensor methods for regression applications," *Chemometrics and Intelligent Laboratory Systems*, vol. 152, pp. 69–79, Mar. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169743915003263

[3] Y. Jiang, S. Yin, J. Dong, and O. Kaynak, “A review on soft sensors for monitoring, control, and optimization of industrial processes,” *IEEE Sensors Journal*, vol. 21, no. 11, pp. 12 868–12 881, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9235582

[4] J. G. Gallareta, C. González-Menorca, P. Muñoz, and M. V. Vasic, “Advancements in Soft-Sensor Technologies for Quality Control in Process Manufacturing: A Review,” *IEEE Sensors Journal*, vol. 25, no. 9, pp. 14 575–14 588, May 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10931829

[5] “European Critical Raw Materials Act,” accessed on 2026-01-23. [Online]. Available: https://single-market-economy.ec.europa.eu/publications/european-critical-raw-materials-act_en

[6] E. Wan and R. Van Der Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, Oct. 2000, pp. 153–158. [Online]. Available: https://ieeexplore.ieee.org/document/882463

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ser. Adaptive Computation and Machine Learning series, F. Bach, Ed. Cambridge, MA, USA: MIT Press, Nov. 2016.

[8] C. Shang, F. Yang, D. Huang, and W. Lyu, “Data-driven soft sensor development based on deep learning technique,” *Journal of Process Control*, vol. 24, no. 3, pp. 223–233, Mar. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959152414000365

[9] W. Ke, D. Huang, F. Yang, and Y. Jiang, “Soft sensor development and applications based on LSTM in deep neural networks,” in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov. 2017, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8280954

[10] X. Wang and H. Liu, “Soft sensor based on stacked auto-encoder deep neural network for air preheater rotor deformation prediction,” *Advanced Engineering Informatics*, vol. 36, pp. 112–119, Apr. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034616302865

[11] X. Yuan, S. Qi, Y. A. W. Shardt, Y. Wang, C. Yang, and W. Gui, “Soft sensor model for dynamic processes based on multichannel convolutional neural network,” *Chemometrics and Intelligent Laboratory Systems*, vol. 203, p. 104050, Aug. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169743920300526

[12] Q. Sun and Z. Ge, “Deep Learning for Industrial KPI Prediction: When Ensemble Learning Meets Semi-Supervised Data,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 260–269, Jan. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/8970515

[13] F. Chollet *et al.*, “Keras,” https://keras.io, 2015.

[14] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: a system for large-scale machine learning,” in *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, ser. OSDI’16. USA: USENIX Association, Nov. 2016, pp. 265–283.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[16] J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001. [Online]. Available: https://www.jstor.org/stable/2699986

[17] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6795963

[18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” Dec. 2014, arXiv:1412.3555 [cs]. [Online]. Available: http://arxiv.org/abs/1412.3555

[19] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: ACM, 2016, pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785