Kabugo, James; Jämsä-Jounela, Sirkka-Liisa; Schiemann, Robert; Binder, Christian

Process Monitoring Platform based on Industry 4.0 tools: a waste-to-energy plant case study

Published in:
4th Conference on Control and Fault Tolerant Systems (SysTol)

DOI:
10.1109/SYSTOL.2019.8864766

Published: 14/10/2019

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Kabugo, J., Jämsä-Jounela, S.-L., Schiemann, R., & Binder, C. (2019). Process Monitoring Platform based on Industry 4.0 tools: a waste-to-energy plant case study. In *4th Conference on Control and Fault Tolerant Systems (SysTol)* (pp. 264-269). (Conference on Control and Fault Tolerant Systems (SysTol)). IEEE. https://doi.org/10.1109/SYSTOL.2019.8864766

2019 4th Conference on Control and Fault Tolerant Systems (SysTol)
Casablanca, Morocco, September 18-20, 2019

Process Monitoring Platform based on Industry 4.0 tools: a waste-to-energy plant case study

James Clovis Kabugo
Department of Chemical and Metallurgical Engineering
Aalto University
Espoo, Finland
james.kabugo@aalto.fi

Sirkka-Liisa Jämsä-Jounela
Department of Chemical and Metallurgical Engineering
Aalto University
Espoo, Finland
sirkka-liisa.jamsa-jounela@aalto.fi

Robert Schiemann
Outotec GmbH
Frankfurt, Germany
robert.schiemann@outotec.com

Christian Binder
Outotec GmbH
Frankfurt, Germany
Christian.binder@outotec.com

***Abstract*— This work presents a process data analytics platform built around the concept of industry 4.0. The platform utilizes the state-of-the-art industry internet of things (IIoT) platforms, machine learning (ML) algorithms and big-data software tools. The industrial applicability of the platform was demonstrated by the development of soft sensors for use in a waste-to-energy (WTE) plant. In the case study, the work studied data-driven soft sensors to predict syngas heating value and hot flue gas temperature. From data-driven models, the neural network based nonlinear autoregressive with external input (NARX) model demonstrated better performance in prediction of both syngas heating value and flue gas temperature in a WTE process.**

Keywords—Industrial internet of things, machine learning, waste-to-energy, soft sensor

I. INTRODUCTION

The global competition in industrial manufacturing, for instance to increase productivity, product quality [1], process safety in addition to economic and environmental sustainability has led to the development of modern process monitoring and data analytics systems. The current developments concerning industrial internet of things (IIoT) technologies, machine learning (ML) algorithms and big-data availability provide platforms for the realization of sophisticated process data analytics. There are several cloud computing platforms available for use in industrial internet of things and big-data analytics. Key players include cloud services providers (like Microsoft Azure, Amazon Web Services, IBM, Intel etc.), enterprise solution vendors (notably PTC and Oracle), networking companies (such as Cisco and AT&T) and industrial engineering companies (namely General Electric, Siemens and Bosch), among others. A number of these platforms are available under proprietary licenses with a few others being accessible as open source projects.

In predictive data analytics in process industry, soft sensors are invaluable tools for providing insights into the state of process operations especially in cases where the direct measurement of key process variables is extremely difficult, nearly impossible or even unreliable [2]. For example in waste-to-energy plants (WTE), a number of researchers have reported studies concerning the use of data-driven soft sensors as alternative tools for the prediction of critical process variables such as the calorific value of biomass solid waste, syngas composition, NO_x emissions and oxygen content in flue gas [3].

The rest of this paper is organized into five more Chapters. The proceeding Chapter briefly reviews the software platforms for IIoT, ML algorithms and big data analytics in relation to modern process automation. Chapter III introduces the WTE industrial use case. Chapters IV and V discusses the selected data-driven methods and development of data-based soft sensors for a WTE plant respectively. The final Chapter provides a summary of key findings of the current work.

The work contributes to the existing knowledge of process data analytics in modern process automation systems by emphasizing the use of readily available open source or proprietary software tools to develop sophisticated data analytics platforms to support academic research and for use in industrial process applications.

II. PROPOSED PROCESS DATA ANALYTICS PLATFORM

Currently, a number of hardware and software tools purposely for handling process data analytics are available from industrial automation vendors and other computing services vendors. These tools are increasingly being used to develop data analytics platforms within process automation.

A. Platform Description

The platform presented here in Fig. 1, follows the general data-based process monitoring procedures, which include data acquisition, data pre-processing, model design and model maintenance. In industrial processes, data acquisition can be accomplished through a number of interfaces for instance, OPC UA, OPC, Modbus and various network protocols like MQTT, CoAP, REST and HTTP [4]. Currently various IoT platforms can be implemented for data collection from industrial devices notably through edge computing and IoT cloud getaways. The choice of a data acquisition interface also influences the possibility of real-time process monitoring.

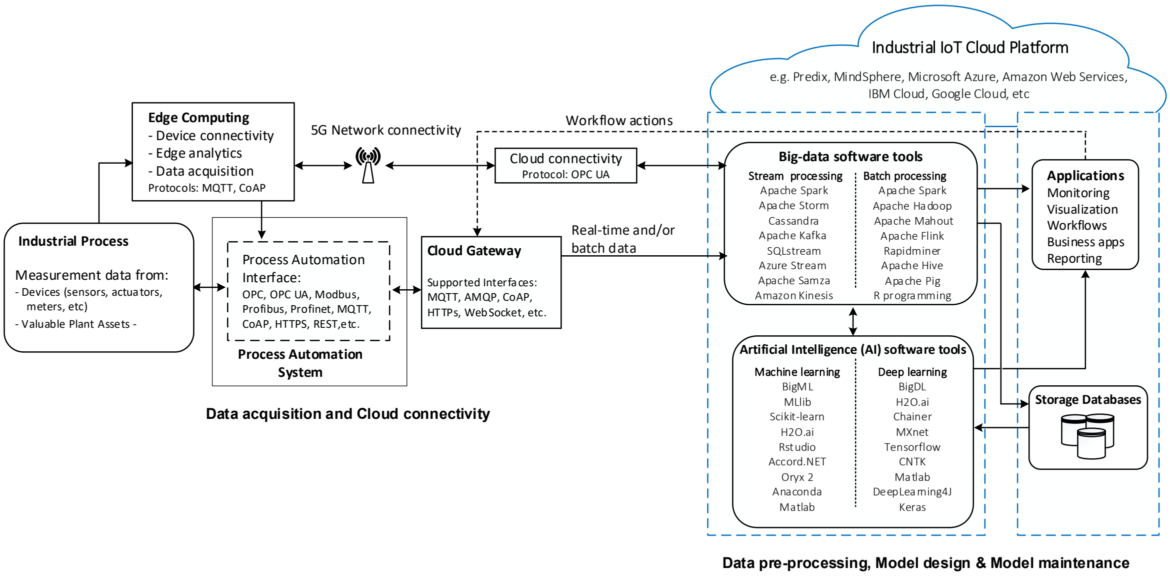

Fig. 1. A detailed process monitoring platform highlighting the use of edge and cloud computing technologies in industrial process automation and data analytics

Thus, in cases of high latencies in data acquisition, offline or batch data analysis is favored. Statistical or ML methods are normally employed during data pre-processing, model design and model maintenance. The choice of the appropriate method in each case depends on the task, among other technicalities.

In addition to the existing process monitoring platforms within process automation systems, modern technologies including edge and IoT cloud computing can be utilized for process data analytics. Through the use of IoT cloud gateways, streaming data or historical data is ingested into the industrial IoT cloud platform. Industrial IoT vendors normally provide IoT cloud getaway connections. For instance, Siemens's MindSphere [5] offers MindConnect Nano and MindConnect IoT 2000 getaways, which ensure secure information sharing. Moreover, with edge computing, the conventional plant automation architecture can be by-passed and the plant devices are directly connected to industrial IoT cloud from the edge by for instance using a 5G communication network.

B. *Industrial IoT Cloud Platforms and Edge Computing*

Nowadays, several commercial grade industrial IoT platforms suitable for use in industrial automation are available, notable examples are Predix, MindSphere [5] and Sentience cloud platforms as a service. In addition, there many multi-purpose IoT cloud platforms, which are available from cloud service providers, for instance, Amazon web services, Microsoft Azure, Google cloud and IBM's Watson IoT. Most of these IoT cloud platforms offer distributed computing, big-data analytics, data and device management tools, machine-to-machine (M2M) interaction and many other functionalities. The connectivity of devices to IoT cloud platforms can be achieved in different ways, for instance by means of 'plug and play' [5], use of open communications standards for industrial automation notably the OPC Unified Architecture (OPC UA) and network protocols like MQTT. Cloud computing platforms also provides large and affordable data storage capacity, flexibility to client demands plus offering highly scalable computing power.

Mobile edge computing (MEC) is among the currently available edge computing technologies within the industrial IoT. The edge interface offers real-time data processing including data filtering, basic data analytics, in addition to machine-to-machine communication, all at the plant premises and near the data sources. This means that fast process monitoring can be achieved through for example anomaly detection models, among others.

C. *Machine Learning and Big-data Software Tools*

For data analytics, there several machine learning software tools available both commercial and open source software. Among the leading proprietary machine learning as a service (MLaaS) include Amazon machine learning (Amazon ML), Microsoft Azure machine learning, Google machine learning and IBM Watson machine learning. Often machine learning tools are accessible through respective cloud platforms or under third party cloud application services. They offer highly scalable environments and a variety of machine learning algorithms for data pre-processing, dimensionality reduction, predictive data analytics and plenty of other functionalities. Moreover, most of the commercial machine learning frameworks also provide deep learning libraries and big-data computing software tools like Apache Spark, Hadoop, etc. In addition, there are several machine learning software tools, libraries and frameworks, which are freely accessible under open source licenses. Also, these can be implemented via either through the cloud environment or on premises. Examples of open source machine learning software tools are Apache Spark MLlib, Scikit-learn, TensorFlow, H2O.ai, BigML, Accord.NET, Apache SystemML, Apache Mahout, Oryx 2, just to mention but a few. There are also many available open source deep learning frame works and libraries, which offer different deep learning models for big-data analytics. Notable examples are H2O.ai, Tensorflow, Chainer, MXnet, Keras, BigDL, Microsoft Cognitive toolkit (CNTK) and many more. Furthermore, some proprietary ML vendors offer big-data stream analytics platforms for example Azure Stream Analytics and Amazon's Kinesis. With a variety of ML algorithms, soft sensing models among others can be developed and implemented in process monitoring

III. WASTE-TO-ENERGY PLANT INDUSTRIAL USE CASE

A waste-to-energy plant was studied here as an example for application of the above proposed monitoring platform. Waste-to-energy plants usually treat different kinds of solid waste materials, which may include municipal solid waste, industrial sludge, sewage sludge, agricultural waste, wood waste, plastic waste and so forth. The heterogeneity of solid fuel, environmental constraints, profitability, among other limitations, make the operations of the WTE plant considerably complex. Therefore, to achieve better plant operation efficiencies, continuous process monitoring is very important. In this case study, data-based soft sensors applicable in the prediction of syngas heating value and hot flue gas temperature were developed.

A. Process Description

In this process, solid waste fuel is fed into the Outotec Advanced Staged Gasifier where it is combusted to produce heat. The heat generated is used to produce steam for electric power generation. The operations of this particular WTE plant are summarized in Fig. 2. The main process unit is the Outotec Advanced Staged Gasifier, which is divided into two parts, the lower and upper sections. In the lower section, gasification takes place in a bubbling fluidized bed where a sand bed and solid waste particles are fluidized by air. The gasification stage produces syngas, which is directed to the upper section for combustion. Any metallic components and large-sized particulates are removed by discharging part of the bed material through the bottom cone of the fluidized bed. However, after cleaning, the bed material is recycled back to the gasifier. In the upper section of the Outotec Advanced Staged Gasifier, the combustion of syngas is carried out in the presence of air. Reagents for NOx reduction, such as ammonia are added in the upper section. Such reagents react with NOx compounds to form harmless nitrogen and water. The flue gas exits the gasifier typically at a temperature of 930ºC. Heat is recovered from the hot flue gas stream using the boiler to produce super-heated steam for electric power generation. After the boiler, flue gas undergoes a series of cleaning stages as well as further heat recovery measures before it is discharged to the environment. At the same time during flue gas cleaning, subsequent solid particulates in form of ash, are collected for disposal or for further treatment and use.

B. Syngas Heating Value and Flue Gas Temperature

Two process variables, syngas heating value and hot flue gas temperature were selected for study. The study was aimed at developing data-based soft sensors for prediction of the syngas heating value and hot flue gas temperature for a WTE plant. Under normal plant operations, the heating value of syngas is difficult to quantify. A data-driven soft sensor in that case would be very useful for the online prediction of syngas heating value. On the other hand, the combustion chamber outlet temperature affects various downstream process

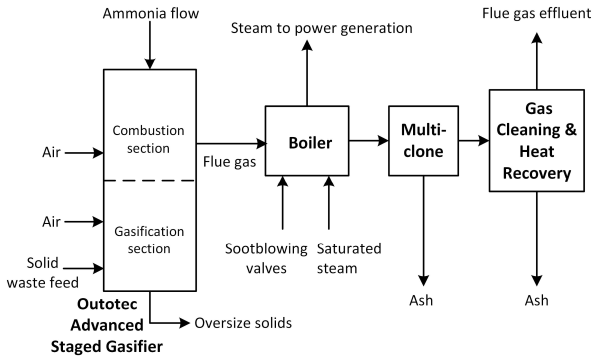

Fig. 2. A simplified process scheme for a WTE plant showing relevant streams and variables studied in this work

operations including turbine power output and emissions of harmful gaseous compounds. Therefore, a good prediction model for the flue gas temperature at the outlet of the combustion chamber would also be useful in the process control strategies, process optimization and process monitoring. For instance, a soft sensor for flue gas temperature would be applicable in fault detection if the temperature sensor and soft sensor outputs disagree too much and hence create alarms.

IV. DATA-DRIVEN MODELS

The list of machine learning options, which could be applicable in this case is rather long. However, some of the notable methods according to literature include variants of linear regression methods, support vector regression (SVR), Bayesian networks regression, variants of artificial neural networks (ANNs), decision trees regression, random forest regression, k-nearest neighbor (K-NN), principal component analysis (PCA) based methods, adaptive boosting (AdaBoost) method and so on [6]. Again, most of these machine learning algorithms are available through several ML software frameworks and libraries with examples presented in Fig. 1. For instance, Ahmed et al. [7] predicted and forecasted energy consumption for a smart grid system using support vector regression, linear regression and artificial neural network methods with the help of the IBM Cloud machine learning tools. For this work, a dynamic neural network based nonlinear autoregressive model with external input (NN-NARX) was employed. It's performance was compared with a linear and static model based on multivariable linear regression (MLR). The choice of the NN-NARX model was also motivated by it's ability to solve time series forecasting.

A. Neural Network-Nonlinear Autoregressive Exogenous Model

Equation error models similar to the NN-NARX model are widely known in the prediction and control applications. The NARX model estimates the observed response based on the previous inputs and output variables as described in (1).

$$y(t) = f[y(t-1),...,y(t-n_a),...,u(t-n_k),..\\ .,u(t-n_k-n_b+1)] + e(t) \tag{1}$$

where, $y(t)$ is the output variable at time t, f is the nonlinear function, $u(t)$ is the input variable at time t, n_a is the number of poles, n_b is the number of zeros plus unit, n_k is the delay with respect to the input variable and $e(t)$ is a white noise disturbance. Initially, the nonlinear function f is unknown. However, it is approximated in the training phase of the feedforward neural network model. The neural network NARX model can be implemented using an open loop or a closed loop architecture [8]. The open loop architecture, favors online process monitoring whereas with the closed loop architecture, future process outputs can be estimated offline, for example through process simulation.

Training of the NN-NARX model is often carried out using different types of optimization algorithms notably backpropagation methods. Here, the commonly applied Levenberg-Marquardt (LM) algorithm was utilized. Moreover, in the estimation of the model performance, the mean squared error (MSE) method, which is described in (2) was employed. In which, y_i is the measured output variable, $\hat{y}_i$ is the predicted output variable and n is the number of measurements.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{2}$$

B. Multivariable Linear Regression Model

A multivariable linear regression (MLR) model can be demonstrated as shown in (3) where Y is a n-dimensional response vector, X is a design matrix of predictor variables ($x^{(1)},..., x^{(m)}$) with dimensions of $n \times p$ (and $p = m + 1$), θ is a p-dimensional vector containing estimated parameters and ε is an n-dimensional vector of estimated error. The dimensions n and m correspond to the number of samples and variables respectively. The superscript T denotes the transpose of the respective matrix [9].

$$Y = \theta X + \varepsilon, \text{ where } \theta = \left(X^T X\right)^{-1} X^T Y \tag{3}$$

The performance of the regression model is often assessed based on the estimated coefficient of determination, R^2 and root mean squared error (RMSE) described in (4).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{n}} \tag{4}$$

V. Soft Sensor Development

The data used in the current studies was obtained from the Outotec' process simulator and it represented a period of three months of plant operation with a one minute time interval and equivalent to about 126500 sample data points. In the modeling phase using the neural network model, the data was randomly divided into three parts, for training, validation and testing at 70%, 15% and 15% respectively. For the MLR model, data was split manually into two sets, for model training and testing at 80% and 20% respectively. All the work, including data pre-processing and modeling was conducted using Matlab 2018a software. The neural network toolbox in Matlab was utilized in the implementation of the NN-NARX model.

A. Data Pre-processing and Variable Selection

For the detection and removal of outliers, the Hampel identifier method was implemented. In the next step, data filtering was done using a median filter of the seventh order. This was found appropriate in terms of minimizing the noise in the data and avoiding the loss of important information from the data. Due to varying data scales, the data was scaled before further data analysis. The selection of predictor variables was done based on data correlation analysis and process knowledge. The correlations between the independent and dependent variables were examined from the correlation matrix and highly correlated independent variables against the target dependent variables were selected for further analysis. Process knowledge was used to eliminate independent variables, which were strongly correlated with the dependent variables but already known as unsuitable predictor variables according to process operations. In addition to the original independent variables, calculated variables, which exhibited stronger correlation behavior towards the output variables, were also included in the development of the respective prediction model. The original and computed variables used in the data-based prediction models are shown in Table I.

TABLE I.
List of variables and their descriptions

Variable	Description
x_1	Syngas temperature
x_2	Fluidized bed inlet air volumetric flow
x_3	Combustion chamber inlet air volumetric flow
x_4	Fluidized bed air valve opening
x_5	Combustion chamber air valve opening
x_6	Product of the combustion chamber air volumetric flow and valve opening
x_7	Product of the fluidized bed air volumetric flow and valve opening
x_8	Combustion chamber inlet air pressure
x_9	Solid fuel feed valve opening
x_{10}	Product of the combustion chamber air volumetric flow and pressure
x_{11}	Boiler feed water
y_1	Syngas heating value
y_2	Combustion chamber outlet flue gas temperature

B. Prediction of Syngas Heating Value

In the prediction of syngas heating value, input variables which included x_1, x_2, x_3, x_4, x_5, x_6 and x_7, from Table I were retained based on the correlation studies. After tuning the neural network NARX model as illustrated in Table II, a neural network architecture with 10 neurons and delays of 7 min was adopted. The selection was based on the observed mean squared error values for each scenario and taking into account, other performance measures such as the time required in the model training, error autocorrelation as well as the input-error cross-correlation behaviors. For instance, it was ensured that the results obtained from the error autocorrelation and the input-error cross-correlation functions in Matlab were within the recommended 95% confidence limits. Otherwise, the model was re-trained until the acceptable results were obtained. Fig. 3 shows the prediction results of syngas heating value from the open-loop NN-

NARX model. Here, it can be seen that the model predicted syngas heating value with high accuracy. The observed average values of MSE and correlation coefficient, R^2 after model testing were 0.0021 and 0.988 respectively. On the other hand, application of the MLR method yielded the results shown in Fig. 4. The MLR model showed a correlation coefficient R^2 of 0.868 and a RMSE value of 17.40 on a test dataset. Comparing the test results in Fig. 3 and Fig. 4, it's visibly clear that the NN-NARX demonstrated better prediction performance for syngas heating value than the MLR model.

TABLE II.
Performance results of the neural network NARX model for selected time delays and the number of nodes within the hidden layer (after five parallel experimental runs)

No. of hidden layer nodes	Time delays, min	Mean squared error (MSE)		Regression coefficient, R^2		Average training time, min
		Mean value	*Standard deviation*	*Mean value*	*Standard deviation*	
7	7	0.00209	3.19E-05	0.9879	2.76E-04	1.52
10	**7**	**0.00205**	**1.67E-05**	**0.9881**	**3.07E-04**	**2.35**
14	7	0.00202	1.76E-05	0.9880	1.82E-04	4.23
10	5	0.00207	5.16E-05	0.9878	2.54E-04	2.61
10	10	0.00203	9.57E-06	0.9880	1.40E-04	3.75

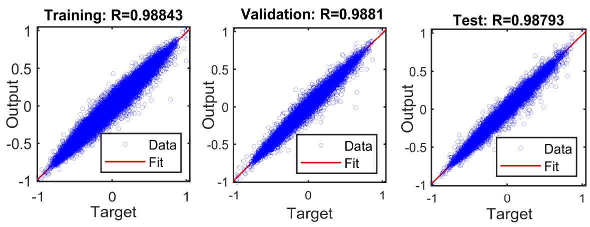

Fig. 3. Prediction of syngas heating value using a neural network NARX model

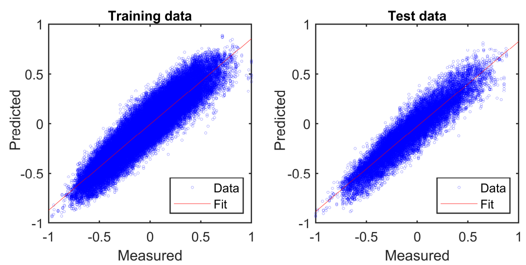

Fig. 4. Prediction of syngas heating value using a multivariable linear regression model.

The neural network NARX model was also applied in forecasting (multistep ahead prediction) of syngas heating value. During plant operations, the measurement of syngas heating value is done intermittently and hence time series forecasting would be useful in monitoring the performance of the combustion process continuously. Forecasting was carried out by switching the trained open-loop mode neural network into a closed loop mode. With a set of input data for only the predictor variables, the closed-loop neural network was used to predict the corresponding output values. The predicted output was compared with the target output, which in this case

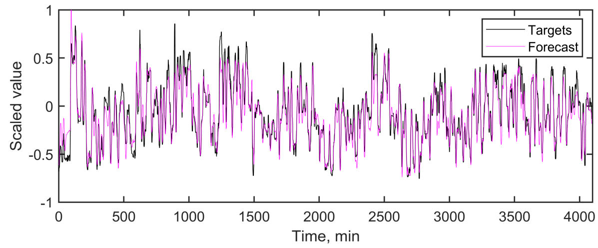

Fig. 5. Forecasting of syngas heating value using a neural network NARX model.

was already known. Fig. 5 shows the forecasting results of syngas heating value using the closed-loop NN-NARX model. Again the model performed considerably well in forecasting the heating value of syngas for over two days' operation with an average value of MSE of 0.0120 and a regression coefficient R^2 as high as 0.931 observed.

C. *Prediction of Flue Gas Temperature*

The temperature of flue gas exiting the combustion chamber was modeled using predictor variables, which included x_2, x_3, x_4, x_8, x_9, x_{10} and x_{11} from Table I. Similarly, to the previous case, the predictor variables were selected based on their significant correlation with the output variable. Also here, the open-loop NN-NARX model performed quite well in the prediction of hot flue gas temperature as illustrated in Fig. 6. In this case, the observed average values of the MSE and corresponding correlation coefficient, R^2 after model testing were 6.23E-04 and 0.998 respectively. Similarly, the MLR model was studied using the same variables and it's results are presented in Fig. 7. The MLR method still performed fairly well, with a regression coefficient of 0.89 and RMSE of 14.07 observed. However, again the NN-NARX model demonstrated superior performance between the two studied models on the same predictor variables.

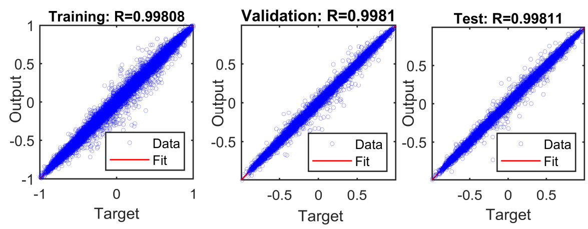

Fig. 6. Prediction of hot flue gas temperature using a neural network NARX model.

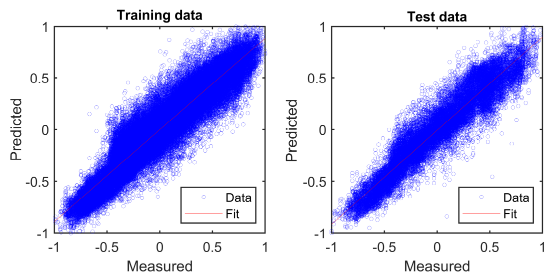

Fig. 7. Prediction of hot flue gas temperature using a multivariable regression model.

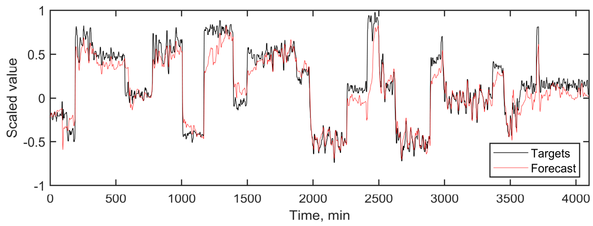

Fig. 8. Forecasting of hot flue gas temperature using the neural network NARX model.

Furthermore, a closed loop model was applied and a new dataset of predictor variables was used to forecast the respective output values. The results are shown in Fig. 8, where the forecasted flue gas temperature is compared with the target output values, which were already known prior to forecasting. The NN-NARX forecasted hot flue gas temperature fairly well for the studied future period of about two days. The average values of the observed MSE and correlation coefficient R^2 were 0.0204 and 0.936 respectively.

D. Sensitivity Analysis

Sensitivity analysis was performed in order to establish the most important independent variables with respect to the applied data-driven prediction model. A stepwise technique [10] was utilized in the backward manner. To quantify model's sensitivity to each variable, the difference in RMSE observed between the model with all input variables and for a model corresponding to a particular eliminated independent variable, was calculated. The error difference was expressed as a fraction according to (5).

$$\varepsilon_d = \left(\varepsilon_{n-i} - \varepsilon_n\right) / \varepsilon_n \tag{5}$$

In which, ε_d is the error difference as a fraction, ε_{n-i} represents the RMSE when input variable i is eliminated from the model and ε_n is the RMSE corresponding to the model with all the originally selected n input variables. The open-loop NN-NARX model was re-trained in each case at least five times after the initial run. The results are presented in Fig. 9 where the relative importance of each independent variable is described by the percent error difference calculated from (5). In the prediction of syngas heating value Fig. 9a, variables x_1 and x_4 were observed to be less important in the design of MLR model and thus one of these variables can be removed from each model without significant loss of model performance. Whereas variables x_3 and x_5 showed the most significant contribution to the performance of the MLR model. However, for the NN-NARX model, variables x_3 and x_7 were found to be far less significant. Otherwise, each variable showed moderately small individual contributions within the model. From the modeling of flue gas temperature in Fig. 9b, variables x_2, x_4 and x_{10} showed low to no importance within the models. Variables x_3 and x_{11} showed strong influence in the MLR model but far less contribution within the NN-NARX model. However, in all models, variable x_9 was observed to play a significant role.

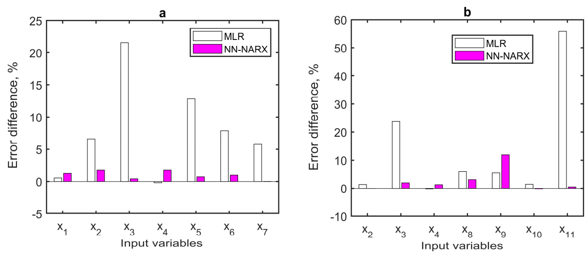

Fig. 9. Relative importance of the independent variables within the prediction models: (a) for syngas heating value and (b) for hot flue gas temperature.

VI. Conclusions

Industrial IoT software platforms, machine learning frameworks and libraries and big-data tools applicable in developing and implementing process data analytics in modern process automation systems were introduced. Moreover, a process monitoring platform scenario was developed based on these tools. With such environment, different data-driven models can be realized. The application of the platform was demonstrated by developing data-driven soft sensors, which can be employed in monitoring syngas heating value and hot flue gas temperature in a WTE plant. The dynamic neural network based NARX model showed high prediction performance in all cases. The model also showed low sensitivity towards the selected independent variables.

References

[1] A. J. Trappey, C. V. Trappey, U. H. Govindarajan, J. J. Sun and A. C. Chuang, "A Review of Technology Standards and Patent Portfolios for Enabling Cyber-Physical Systems in Advanced Manufacturing," IEEE Access, vol. 4, pp. 7356-7382, 2016.

[2] 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, pp. 223-233, 2014.

[3] G.-Q. Li, X.-B. Qi, K. C. Chan and B. Chen, "Deep Bidirectional Learning Machine for Predicting NOx Emissions and Boiler Efficiency from a Coal-Fired Boiler," Energy&Fuels, vol. 31, pp. 11471-11480, 2017.

[4] P. O'Donovan, K. Leahy, K. Bruton and D. T. J. O'Sullivan, "An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities," Journal of Big Data, vol.2, no. 25, pp. 1-26, 2015.

[5] Siemens, "Introduction to MindSphere Predictive Learning," [Online]. Available:https://documentation.mindsphere.io/resources/html/predictive-learning/en-US/MDSPPredictiveHelp.htm#t=Introduction_to_Predictive_Learning.htm. [Accessed 31st October 2018].

[6] P. Kumar, T. Amgoth and C. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," Information Fusion, vol. 49, p. 1–25, 2019

[7] A. Ahmed, K. Arab, Z. Bouida and M. Ibnkahla, "Data Communication and Analytics for Smart Grid Systems," in 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20-24 May 2018.

[8] A. Salah, L. Hanel, M. Beirow and G. Scheffknecht, "Modelling SER Biomass Gasification Using Dynamic Neural Networks," in 26th European Symposium on Computer Aided Process Engineering, Portorož, Slovenia , June 12th -15th, 2016.

[9] S. H. Brown, "Multiple Linear Regression Analysis: A Matrix Approach with Matlab," Alabama Journal of Mathematics, pp. 1-3, 2009.

[10] M. Cao, N. F. Alkayem, L. Pan and D. Novák, "Advanced Methods in Neural Networks-Based Sensitivity Analysis with their Applications in Civil Engineering," in Artificial Neural Networks: Models and Applications, Rijeka, Croatia, IntechOpen, 2016, pp. 335-353.