Prediction and Explanation Models for Hot Metal Temperature, Silicon Concentration, and Cooling Capacity in Ironmaking Blast Furnaces

Edwin Lughofer,* Robert Pollak, Christoph Feilmayr, Magdalena Schatzl, and Susanne Saminger-Platz

Herein, the establishment of data-driven prediction and explanation models for three essential process variables in ironmaking blast furnace processes, namely hot metal temperature, silicon concentration, and cooling capacity is demonstrated. Aside a reliable prediction quality of the models with sufficient prediction horizon, an additional main goal has been to establish interpretable and revealing models. To support (linguistic) interpretability, the main focus has been set on the extraction of rule-based models from a large database collected at a particular blast furnace process located at the partner company’s site. Due to expected uncertainty in the data, due to, e.g., measurement noise, fuzzy systems are an adequate architectural choice for achieving models in a robust rule-based form. For a fully automatic training of the predictive fuzzy systems new feature ranking methods have been performed on the one side and a special granular rule extraction procedure on the other side. Testing the obtained models on two separate test datasets from consecutive years shows a stable prediction performance without any error drifts and a higher performance than other related machine learning methods including deep neural networks, and others. Moreover, the final models, turned out to have maximal 4–5 inputs and just a couple of rules and allowed to gain new insights into the processes.

1. Introduction

1.1. Motivation and State-of-the-Art

Iron and steel industry faces increasing challenges due to efforts in terms of carbon input reduction, global competition, and considerable over capacity of production facilities. Thus, the optimization of production costs—in addition to securing highest product quality—is essential for economic survival. Closed-loop process models for proper prediction of future system states usually can reduce the costs significantly, see the study by Pal et al. The blast furnace process is on the one hand highly energy intensive, on the other hand implemented in huge production units—biggest furnaces exceed 6000 m³ of inner volume. Therefore, every small optimization of the process leads to respectable energy or raw materials savings in absolute numbers. One way of optimizing is the automated regulation of reducing agent input (like coke, pulverized coal injection, waste plastic, natural gas), to stabilize the thermal state of the furnace. Usually, this automation procedure considers the actually measured hot metal temperature. If it would be possible to reliably predict the future development of this temperature for the coming hours, the control of the thermal state could be significantly improved. The silicon concentration in the hot metal is also strongly influenced by the inner furnace temperature level. In contrast to the temperature, the concentration is not measured continuously, but pointwisely and is not really depending on the currently predominating slag flow at the tap hole. Moreover, the value of the silicon concentration is available before the end of the tapping, thus the prediction could be done earlier. The cooling capacity, which is the amount of energy which is removed by the cooling water in single parts of the blast furnace per time, gives important insights concerning the stability and homogeneity of the gas flow within the packed bed of burden and coke. During unstable, disadvantaged operating conditions the gas flow at the inner walls increases significantly and leads to high cooling capacities. A quantification of emerging unstable conditions would help the process team to take countermeasure.
Due to the lack of thermodynamic and kinetic data for very high temperatures and the complex dependencies between internal system variables and the three targets under examination, it is impossible to establish analytical models based on physical and chemical reactions backgrounds only. Expert systems have been established as promising alternative in the past for blast furnace modeling purposes (ranging back to the end of 1980s and beginning of 1990s), see several pioneering works in the studies by Amano et al., Iida et al., and Tianjun et al. Therein, knowledge is coded based on experience by experts working with the process since several years and thus they allow explainable insights into the process and (predicted) states. In the study by Amano et al., the artificial and logical intelligence system (ALIS) for blast furnace operation embeds expert rules for a proper supervision of the process. The expert rules provide an interim hypothesis for each item under supervision, such as the gas flow distribution or the furnace heat level, whereas a final decision of proper action is made based on the results of the interim hypothesis judgment. In the study by Iida et al., expert rules are coded in a rule book to perform an analysis of the current process situation, a recognition of phenomena and the determination of proper actions. Therefore, several process variables such as hot metal temperature or heat consumption are used, but no future predictions of these variables are made for early reaction purposes. Similar considerations go to the early approach in the study by Tianjun et al., where expert rules are used in four different blocks for abnormal status diagnosis and operation guides, once such occurrences have happened. In the study by Otsuka et al., a combination of an analytical mathematical model and several expert rules has been achieved for establishing a hybrid control system, which is able to reduce energy costs and to stabilize the furnace operation.

The collection and coding of expert knowledge is typically achieved fully manually and within several interviews and discussions, and thus very time consuming and furthermore often affected by human uncertainty, subjectiveness, and/or vagueness issues. Therefore, our aim was to use data-driven modeling methods, which are i) able to establish prediction and explanation models more or less autonomously, and this solely based on (objective) historic data collected during the process, and ii) also allow interpretable insights into the process.

Several data-driven approaches have been suggested in the past for blast furnace modeling and process state evaluation issues with early works in the studies by Saxen and coworkers, and further developments in the study by Saxen and Pettersson, the latter two focusing on the prediction of hot metal temperature and silicon content. In the study by Ostermark and Saxen, a statistical approach based on vectorized autoregressive models with exogenous inputs (VARMAX models) has been developed to predict the hot metal temperature and the silicon content for the next process stage (so, no direct longer-term predictions are made). It uses three inputs based on correlation analysis and establishes global linear models in a closed functional form (including time-lagged variables), achieving explainable insights on a global view, but not piecewise for partial local operating conditions and/or system states. In the study by Saxen and Pettersson, a feed-forward neural network (NN) with sigmoid activation functions and a flat architecture (single layer) is applied for the purpose of predicting the silicon content 1 h ahead with a past history of up to 8 h; it again results in a global model with some interpretable insights according to expedient combinations of input variables used in the first model layer. Similar considerations go to the approach proposed in the study by Jimenez et al., which applies an NARX modeling structure and time averages of input variables as a kind of prelayer before the actual input layer and further hidden layer, making the model less interpretable.

More recent developments can be found in previous studies. Various data-driven modeling techniques, such as Gaussian process regression, support vector regression, artificial and deep NNs, pattern trees and others have been used in these works to establish models with highest possible prediction accuracy. However, little care has been taken to achieve fully autonomous data-driven models which also allow interpretable and explainable insights into the process, i.e., to gain knowledge about which relations and dependencies among certain system variables are most apparent to explain the target and its phases or to gain knowledge about the extent of the influence of system variables onto the target. Most of the aforementioned data-driven modeling techniques are operating as full black boxes, which do not allow any explanation leeway. An exception is provided in the study by Zhang et al., where an ensemble of pattern trees is learnt, where the importance of single variables is measured by an extended out-of-bag error measure. However, insights into structural relation level are hard to gain, because of the ensemble nature, which may include a few dozens of pattern trees, each one with a different outlook as generated from various bags, following the classical bagging technique based on bootstrap replications. Thus, no unique single model, which could be well interpreted and confirmed as physically reliable from an expert’s viewpoint, is demonstrated there. Moreover, explicit variable selection, necessary for achieving compact and transparent models also in the case of a larger range of possible input variables, was loosely carried out using simple covariance and correlation measures as done in the studies by Fontes et al. and Ostermark and Saxen. Finally, and to the best of our knowledge, explanatory models for the cooling capacity were not established or published so far at all.

1.2. Our Approach

To tackle the aforementioned restrictions, we established data-driven prediction and explanation models based on a fuzzy systems architecture. These models are able to express the implicitly contained relationship between system variables by antecedent-consequent statements, often referred to as IF–THEN rules. These rules are linguistically readable and thus greatly transparent and well understandable for experts; furthermore, they provide local insights into the process behavior within different partial regions arising from process states and operating conditions—we will provide concrete rule examples in Section 5. Due to the time dynamic nature of the process, we also used time lags of all system variables up to a time frame of 6–8 h to predict the hot metal temperature, the silicon concentration, and the cooling capacity with a prediction horizon of about 140 min. This is because the residence time of burden in the analyzed blast furnace is 6–8 h. Hot metal temperature of the past (at least 3 h before the prediction time) is known to the prediction
algorithm. So, any undetected accumulation of enthalpy within the furnace vessel can be excluded.

The original input domain with a significant number of variables (up to 64) is even further enlarged by considering up to 130 time lags for each of the inputs in the case of modeling the hot metal temperature and silicon concentration (resulting in a look back of 10 h) and up to 96 time lags for each of the inputs in the case of modeling the cooling capacity (resulting in a look back of 8 h)—see Section 3.1 for a more detailed explanation. To train the fuzzy systems robustly without suffering from the huge dimensionality of the final learning problem, we developed a feature selection algorithm which has synergies with the classical forward selection (FS) mechanism,[19] but extends it by successive orthogonalization of the dependent feature. The aim for doing so is twofold: on the one hand a reduction of computation time shall be achieved; on the other hand, variables best explaining a target in a nonlinear (fuzzy system) sense shall be detected in this way—note that the original FS method relies solely on linear predictors and measures (for more details, see Section 3.2).

Once the features with their best lags are properly selected, our learning process conducts a combination of i) single-pass rule evolution and adaptive antecedent learning and ii) of a new robust learning scheme of the rules’ consequent parameters. As explained in Section 3.3, the rule evolution is based on the generalized smart evolving fuzzy systems approach (GS-EFS)[20] for finding the proper nonlinearity in the process for predicting and explaining the targets, respectively. The learning scheme of the rules’ consequent parameters is based on a total least squares functional (TLS) respecting noise in inputs as well as outputs, which is expected to be apparent in the data from the blast furnace process—see Section 3.3.

The learning approach is evaluated on a long-term dataset collected from the blast furnace process containing samples from 3 consecutive years. Its performance is compared with those of multilayer perceptron NNs (up to five layers) and several other related data-driven methods, as well as with conventional fuzzy systems using axis-parallel rules—see Section 5. The extracted rules will be explicitly listed and its knowledge gaining capabilities explained as well as their expert-based interpretation discussed.

2. Application Scenario

In 2019, a share of more than 70% of the global steel output is produced via the blast furnace-BOF-route (https://www.worldsteel.org/steel-by-topic/statistics/World-Steel-in-Figures.html). In the blast furnace—a type of metallurgical furnace—liquid, carbon saturated iron is produced. Preheated, pressurized air reacts with coke and other reducing agents to form reducing gas. With the heat of the ongoing reactions and the reduction potential of the gas, oxidic iron compounds are smelted and reduced to pig iron. Figure 1 (taken from https://en.wikipedia.org/wiki/Blast_furnace#/media/File:Blast_furnace_NT.PNG) shows the different parts of a blast furnace plant. Therein, solid raw material is charged into the reaction vessel at No. 4 and hot metal is tapped at No. 9.[21] The reactor itself is shown in more detail in Figure 1.

The thermal state of a furnace is usually controlled via the hot metal temperature (process variable “RE_RE_TMP” in the right lower part of Figure 2) and/or the silicon concentration in the iron (process variable “RE_ANA_SI” in the right lower part of Figure 2). Whenever the furnaces cool down, the specific input of reducing agents (e.g., 1 kg coke per kg hot metal) will be increased. Homogenous gas flow within the packed bed of the burden is crucial for a smooth and stable operation. To assess gas flow stability, it is common practice to monitor the cooling capacity of the upper parts of the furnace vessel—of the so-called bosh, belly, and stack. High values of cooling capacity indicate increased hot gas flow at the wall region and thus inhomogeneous gas distribution and suboptimal operation. If it would be possible to predict the appearance of such conditions, the process operator could take countermeasures earlier. The involved process variables are shown in Table 1.
In our application scenario, all analyzed data were recorded at blast furnace 6 of voestalpine Stahl GmbH. It is a nonpressurized furnace with an inner volume of about 1200 m³ and an average production of some 100 t of hot metal per hour.

3. Methodology

3.1. Problem Statement

Assuming a significant number of input variables $\tilde{x} = \{x_1, \ldots, x_J\}$ collected from the blast furnace process, which should be considered for establishing prediction and explanation models of the target variables under examination (hot metal temperature, silicon concentration, and cooling capacity), assuming a prediction horizon of $m$ minutes and a maximum time lag $K$ of the variables to be respected, the goal of the modeling process is to establish a predictor $f$ (for the target variable under examination) in terms of the following functional form

$$\text{target}(t + m) = f(\tilde{x}(t), \tilde{x}(t - 1), \ldots, \tilde{x}(t - K))$$  \hspace{1cm} (1)

where $t$ is the current (present) time instance and $t – K$ represents the time instance which lies $K$ measurement steps in the past (depending on the resolution of the recordings). In our application, the input variables to be respected were selected by expert knowledge in advance to achieve a first domain-knowledge-based dimension reduction, which already may reduce the likelihood of problems due to the curse of dimensionality. The expert-based selection process typically ended up with at least 30 input variables for which up to 130 time lags were to be considered.

When gathering samples from the process with considering a past time line, see Section 4.1, a 3D data matrix for the prediction problems can be spanned up (see also Figure 3). Note that, $N$ denotes the number of samples, $J$ is the number of original input variables, and $K$ the number of time lags to be maximally respected (based on the expert input).

Thereby, in general, $Q$ targets $y_1, \ldots, y_Q$ are predicted (in our case $Q = 3$)—for the sake of simplicity, we neglect the index in the occurrence of target $y$ below, because the models for each of the three targets can be established separately and fully independently. By flattening this matrix into a 2D one by folding the third dimension, we obtain the following time-series-based regression matrix as input to our modeling procedure

$$R = \begin{bmatrix}
    x_1(t_1) & x_1(t_1 - 1) & \cdots & x_1(t_1 - K) & \cdots & x_J(t_1) & x_J(t_1 - 1) & \cdots & x_J(t_1 - K) \\
    x_1(t_2) & x_1(t_2 - 1) & \cdots & x_1(t_2 - K) & \cdots & x_J(t_2) & x_J(t_2 - 1) & \cdots & x_J(t_2 - K) \\
    \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
    x_1(t_N) & x_1(t_N - 1) & \cdots & x_1(t_N - K) & \cdots & x_J(t_N) & x_J(t_N - 1) & \cdots & x_J(t_N - K)
\end{bmatrix}$$  \hspace{1cm} (2)
with \( N \) being the number of available samples at time points \( t_1, \ldots, t_N \); thereby, \( x_J(t_1), x_J(t_1 - 1), \ldots, x_J(t_1 - K) \) can be seen as a time series for variable \( J \), which are amalgamated over all variables. Furthermore, \( \tilde{y} \) is defined as the target vector (the variable under examination to be predicted): \( \tilde{y} = [y(t_1 + m), y(t_2 + m), \ldots, y(t_N + m)]^T \), with \( m \) being the prediction horizon. Self-recursivity, i.e., the integration of past values of the target variable itself for predicting its future trend, has been allowed for establishing the prediction models for hot metal temperature and silicon concentration. In this case, \( y(t_1), y(t_1 - 1), \ldots, y(t_1 - K) \) are added as columns to the \( i \)th row in the regression matrix \( R \) leading to a vectorized autoregressive modeling procedure using exogenous inputs \( (x_1, \ldots, x_J) \), whose resulting model is then termed as VARMAX model.\(^{[24,25]}\)

### 3.2. Dimension Reduction

As already outlined, interpretable models were requested by the company experts for gaining additional insights into the process. Therefore, time-series transformation techniques such as partial least squares regression-based methods as used e.g., in the studies by Lughofer et al.\(^{[26,27]}\) were not a feasible option, because the meaning of the resulting scores in the latent variable space are completely unclear to experts. Therefore, we applied a feature ranking approach, which treats each column in \( R \) (thus one variable with an associated time lag) as one feature and which generates a ranking of these features according to their importance for predicting the target with the chosen prediction horizon \( m \). Taking a subset from the resulting ranked list, i.e., the first \( p \) variables, ends up with a reduced feature space which is then used as an input to the modeling procedure, see Section 3.3. At first glance, this seems to be a classical filter approach,\(^{[28]}\) as dimension reduction is decoupled from the subsequent model learning process, however, we already emphasize the modeling procedure and use the fuzzy systems architecture within the ranking procedure. In this sense, our approach can be seen as a half-filter, half-wraper feature ranking technique.

In particular, for ranking the variables, we use a modified forward selection approach which internally performs an orthogonalization of the dependent after each selection step: this is achieved by subtracting the so-far explained content of the target \( y \) from the original target \( y \) (→ residual) and conducting the next selection step on the residual only—note that, this subtraction operation is in close relation to the vector subtraction step applied in the Gram–Schmidt orthogonalization algorithm\(^{[29]}\) (thus, we call our method “forward selection with orthogonalization”). In each selection round, the variable with the highest correlation to the not yet explained part of \( y \) is selected and added to the ranked list, and so on. According to these univariate selection and subtraction
steps, it is prevented that multiple regressors which are redundant to each other are selected. In this sense, there also robustness assurance issue is integrated, as the final regression matrix spanned by the selected regressors contains hardly any redundancy, thus has usually full rank. In the context of ordinary least squares regression, this typically leads to well-posed solutions when (pseudo-)inverting the regression matrix. The pseudo code explaining the single steps in detail are shown in Algorithm 1.

Algorithm 1 Feature Ranking

1. Input: target vector \( y \), regression matrix \( R \) consisting of \( K \times J \) features (\( J \) system variables, \( K \) time lags), maximal number of inputs \( \text{max}_\text{inp} \); the ordered set of selected features \( L_{\text{sel}} = \{ \} \), the set of all features \( L_{\text{all}} \) (columns in \( R \) as defined in (2)).

2. For \( j \in L_{\text{all}} \), perform univariate regression between feature \( r_j \) and \( y \), elicit correlation between observed and predicted target, Corr \( r_j \).

3. \( \text{sel} = \arg \max_{\text{all} \setminus \text{sel}} | \text{Corr}_r | \).

4. Store selected feature \( r_{\text{sel}} \) into a list of already selected features \( L_{\text{sel}} \) (e.g., \( r_{\text{sel}} \in L_{\text{all}} \) \( \cup \{ r_{\text{sel}} \} \)).

5. Delete selected feature \( r_{\text{sel}} \) from the set of the remaining features \( L_{\text{all}} = L_{\text{all}} \setminus \{ r_{\text{sel}} \} \).

6. Perform a regression on \( L_{\text{sel}} \) obtaining a model \( \hat{f}_{\text{sel}} \).

7. \([\text{Optional: Compute the quality of the current fit} \hat{f}_{\text{sel}}, \text{e.g., as the correlation coefficient between} y \text{and} \hat{f}_{\text{sel}}(R_{\text{red}}); \text{store it into a list of quality values} \text{Qual}_{\text{sel}}].\]

8. Subtract the contribution of all already selected features from \( y \) (orthogonalization of the dependent):

\[
y = y \ominus \hat{f}_{\text{sel}}(R_{\text{red}})
\]

with \( R_{\text{red}} \) the regression matrix including the already selected features in \( L_{\text{sel}} \) as columns, and \( \hat{f}_{\text{sel}}(R_{\text{red}}) \) the predictions on all samples in the regression matrix through the model \( \hat{f}_{\text{sel}} \).

9. If \( |L_{\text{sel}}| \leq \text{max}_\text{inp} \), goto Step 2, otherwise stop.

10. Output: ordered list of most important features \( L_{\text{sel}} \) for approximating target \( y \) (most important first). \([\text{optional: ordered list of quality values} \text{Qual}_{\text{sel}} \text{for successively improved regression fits.}]\]

The main issue here is how the correlation in Step 2 and the regression in Step 5 are carried out. In the classical case, conventional linear correlation and regression are used\(^{19}\)—those variables are selected which explain the target best in a linear predictive approximation sense. When using fuzzy modeling procedures, these variables may not be the best explainable ones in a nonlinear, fuzzy-based approximation sense. Therefore, we use the same regression procedure, as discussed in Section 3.3, in these two steps: in Step 2, this reduces to a univariate fuzzy modeling, as only one input at a time is inspected for correlation (due to the fast greedy nature of our approach); thus, the resulting correlation coefficient can be seen as a fuzzy-based correlation coefficient between the input and the target \( y \); it elicits that variable which explains the (remaining) content of \( y \) best in a fuzzy approximation sense. Thereby, in Steps 2 and 5, the same learning parameter, steering the number of rules, as parameterized for the subsequent fuzzy modeling procedure is used. In this sense, it should be emphasized that the search for the best next variable from a larger list of available ones—up to 10 000 in our case due to the time lags—is done based on univariate views, i.e., single correlations, which makes it much more faster than classical forward selection using the full matrix information of variables selected so far.

Finally, we emphasize the usage of the ranked output list \( L_{\text{sel}} \) of regressors for a reduced input regression matrix into the modeling process. One possibility is to by-measure the subsequently increasing quality of fits when adding newly selected regressors, as indicated in the optional Step 7 in Algorithm 1 (e.g., calculated by \( R^2 \)-measure\(^{10}\)). Then, at the stage where a kind of saturation of the increasing quality trend line is observed, the list can be cut and the first features selected up to the saturation point used as inputs. Figure 4 shows an example of such an increasing quality trend line: the first regressor achieves a fit quality of about 0.5, when adding Regressor 2 the quality increases to about 0.7 and so on—obviously, after five selected regressors, the quality increase starts to saturate, thus an amount of five regressors would be a reliable final choice in this case to be taken as an input to the modeling procedure.

Alternatively, in a full-grid search scenario (see Section 4.2), one may iterate over the full list \( L_{\text{sel}} \), successively adding more and more regressors and carry out modeling steps on them. Then, the ideal number of input regressors can be treated as an additional learning parameter and can be elicited by cross-validation and/or a separate validation dataset. The latter was used in all our experiments.

### 3.3. Training of Predictive Fuzzy Systems

As outlined earlier, interpretable models were requested by the company experts to gain additional insights into the process and for verifying whether the models make sense from a physical and metallurgic point of view (and are not just operating as prediction models with certain precision). Thus, we used the fuzzy systems architecture as feasible choice, because it offers linguistically interpretable rules as partial model components.\(^{31,32}\) Each rule thereby represents a local region in the feature space for which a
partial relation in form of an IF–THEN statement holds; these relations are expressed linguistically such that they can be easily read and understood by experts. We will demonstrate the strength of our approach in Section 5 by providing concrete model examples for hot metal temperature and cooling capacity. To also achieve an accurate model in a predictive regression modeling context, we use the usage of hyperplanes as consequent functionals in the rules, which leads to so-called Takagi–Sugeno (TS) fuzzy systems. These are known to be universal approximators, i.e., being able to model any degree of nonlinearity in the process (data) with sufficiently small error; in this sense, they have been successfully applied to predictive modeling tasks in several industrial applications, see e.g., the study by Pedrycz and Gomide.

The output \( f \) of a TS fuzzy system for an input \( \tilde{x} \) is defined by

\[
\hat{f}(\tilde{x}) = \sum_{i=1}^{p} \Psi_i(\tilde{x}) f_i(\tilde{x}, \bar{w})
\]

(4)

with the normalized rule activation degrees

\[
\Psi_i(\tilde{x}) = \frac{\mu_i(\tilde{x})}{\sum_{j=1}^{p} \mu_j(\tilde{x})}
\]

(5)

and the consequent functionals \( f_i(\tilde{x}, \bar{w}) \)

\[
f_i(\tilde{x}, \bar{w}) = w_{0i} + w_{1i} x_1 + \ldots + w_{pi} x_p
\]

(6)

where \( p \) denotes the number of selected regressors, thus \( p = |L_{sel}| \) and \( w_{0i} \) the intercept of the hyperplane. \( C \) denotes the number of rules, which has to be learnt autonomously from the data, see the following paragraphs, and which induces the granularity of the system and the nonlinearity degree of the model.

In the classical form of TS fuzzy systems, the rule activation degrees \( \mu_i(\tilde{x}) \) are constructed by a \( t \)-norm, which combines fuzzy set membership degrees, usually represented through linguistic terms such as LOW, TALL, INTENSE, etc., through an \( AND \) connection to an overall fulfillment degree. The degrees are thus linguistically interpretable and explainable and indicate how well samples fall into the fuzzy sets of a rule representing a partial local region in the feature space. The most conventional choice is the usage of Gaussian fuzzy sets in combination with the product \( t \)-norm. In the generalized form of TS fuzzy systems, \( \mu_i(\tilde{x}) \) is defined by a multivariate Gaussian representation in the follow form

\[
\mu_i(\tilde{x}) = \exp \left( -\frac{1}{2} (\tilde{x} - \tilde{z}_i)^T \Sigma^{-1}_i (\tilde{x} - \tilde{z}_i) \right)
\]

(7)

with \( \tilde{z}_i \) the center and \( \Sigma_i^{-1} \) the inverse covariance matrix of the \( i \)th rule, which defines its ellipsoidal shape. This achieves more flexibility of the rules in terms of being able to arbitrarily rotate in the feature space—the classical definition always leads to axis-parallel rules, which may restrict their approximation capabilities and compactness, as having been analyzed in detail in the study by Lughofer et al. in the Result section, we will compare both variants in terms of predictive accuracy.

Different to Gaussian process regression (as used in related prior works), our model structure uses Gaussian membership functions to retrieve an input partitioning into fuzzy rules having ellipsoidal shapes according to the local characteristics of the data. As a consequence, each rule represents one partial local model, whereas Gaussian process regression is a nonparametric Bayesian approach by specifying a prior distribution on the linear parameter vector, and relocating probabilities based on evidence (i.e., observed data) using Bayesian inference—this leads to a probability-oriented Gaussian process prior specified by a mean and covariance function leading to one global closed form functional (with a global covariance function, once the kernel is specified). In the case of our fuzzy systems, we end up with \( C \) local covariance matrices describing the local characteristics of the data in a possibilistic manner; the prediction output is thus a weighted amalgamation of the single rule outputs represented by \( f_i \) as formulated in (4).

The training of the rule antecedent space concerns the extraction of an adequate number of rules as well as the learning of the centers and inverse covariance matrices, which appear as nonlinear parameters. For this purpose, we use the so-called generalized smart evolving fuzzy systems (GS-EFS) approach, which is able to extract the number of rules in a fast single pass manner by treating the input regression matrix \( \{R, \tilde{y}\} \) joined with the target vector \( \tilde{y} \) as a (pseudo) stream. Thereby, a statistical-based rule evolution criterion is used which decides when to evolve a new rule based on tolerance regions of prediction intervals in the product space. This leads to a condition whose tolerance threshold is given by a quantile of the \( \chi^2 \) distribution; upon its violation, thus indicating that a new, distinct data cloud appears in the feature space (typically induced by a different operation mode/state), a new rule is added to represent this cloud (mode) properly. The incremental update of the rule centers \( \tilde{z}_i \) is achieved through a weighted version of vector quantization through the expected quantization error as objective function, and the update of the inverse covariance matrices is achieved recursively through the Neumann series trick on matrices. For details including formulas of the antecedent learning procedure, see the study by Lughofer et al.

For training the consequent parameters, the following objective is used in the conventional GS-EFS approach, leading to a weighted least squares estimation for each rule separately, thus also termed as local learning

\[
J_i = \sum_{k=1}^{N} \Psi_i(\tilde{x}(k)) c_i^2(k) \rightarrow \min_{\bar{w}_i}
\]

(8)

where \( c_i^2(k) = (y_i(k) - \hat{y}_i(k))^2 \) is the squared residual between observed and predicted target values, \( \Psi_i(\tilde{x}(k)) \) is the normalized rule membership degree as in (5), and \( N \) the number of training samples. Therefore, only for samples which are close to the \( i \)th rule, the optimization is significantly active; for rules which are further away, \( \Psi_i(\tilde{x}(k)) \) gets close to 0, thus are hardly respected in \( J_i \). In the study by Lughofer (Chapter 2), it has been deeply analyzed that local learning has several advantages over global learning in terms of more stability for matrix inversion, is faster to compute and also achieves a better interpretability of the consequent functions—as snuggling along the real trend of approximation surface, which is needed for our interpretation purposes to be able to provide the local influence of certain input variables onto the output, see also Section 5 for concrete examples.
A stable solution also in the case of higher noise levels of (8) can be achieved by a regularized weighted least squares approach

$$\tilde{w}_i = \left( [\tilde{R} \ \tilde{I}] Q_i [\tilde{R} \ \tilde{I}]^T + \alpha I \right)^{-1} [\tilde{R} \ \tilde{I}]^T \hat{Q}_i \hat{y}$$  \hspace{1cm} (9)

where $\alpha$ serves as regularization parameter, which can be set in advance due to parameter choice techniques, see the study by Bauer and Lukas [43] for a survey; $Q_i$ denotes a diagonal weighting matrix containing the normalized membership degrees in all $N$ training samples given by $Q_i = \text{diag}(\Psi_i(\tilde{x}(1)), \ldots, \Psi_i(\tilde{x}(N)))$, $[\tilde{R} \ \tilde{I}]$ is the regression matrix as in (2) with the inclusion of a vector of ones as last column to estimate the intercept $w_{i0}$.

Although the regularized weighted least squares approach respects the impact of noise on the solution and thus tries to stabilize it, it only handles noise in the target adequately, but does not consider to achieve stable solutions in the case of noise in the inputs, which can usually be assumed to be apparent in our application. To circumvent this bottleneck and thus to obtain a bias-free parameter estimation in the case of noisy targets and inputs, it is necessary to reconstruct both, targets and inputs. Therefore, we use the total least squares (TLS) optimization problem which addresses this aspect by optimizing the following problem (for detailed derivations, see the study by Markovsky and Huffel [44])

$$J_{\text{tot}} = \frac{1}{2N} \left( \sum_{k=1}^{N} \| \tilde{x}_k - \hat{x}_k \|^2 + \sum_{l=1}^{N} \| y_k - \hat{y}_k \|^2 \right)$$  \hspace{1cm} (10)

From a geometric point of view, the optimization of (10) achieves that the normal Euclidean distances between data samples and the hyperplane is minimized, whereas in standard least squares (LS) optimization just the vertical distance is minimized.

In our case of consequent learning in fuzzy systems, we want to apply again a weighted approach to follow the local learning spirit. Hence, we formulate the weighted total least squares (WTLS) optimization problem as

$$J_i = \frac{1}{2N} \sum_{k=1}^{N} \tilde{y}_k \| \tilde{x}_k - \hat{x}_k \|^2 + \sum_{l=1}^{N} \| \tilde{y}_k - \hat{y}_k \|^2$$  \hspace{1cm} (11)

with $\Psi_i(\tilde{x}(k))$ again the normalized membership value of the kth sample in the ith rule. According to the derivation in the study by Markovsky and Huffel [44] for the nonweighted TLS, the following norm needs to be optimized for the weighted version

$$\| \sqrt{\tilde{Q}_i (\tilde{R} - \tilde{\bar{R}})} \|^2$$  \hspace{1cm} (12)

with $\tilde{Q}_i$, the diagonal weight matrix $\tilde{Q}_i = \text{diag}(\Psi_i(\tilde{x}(1)), \ldots, \Psi_i(\tilde{x}(N)))$, and where $\tilde{R}$ denotes the regression matrix in the total least squares sense, i.e., containing the regressors as in (2), joined with the target vector $\tilde{y}$, i.e., $\tilde{R} = [\tilde{y} \ \tilde{R}]$ with $\tilde{R}$ as in (2); $\tilde{\bar{R}}$ denotes the affine linear reconstruction of $\tilde{R}$ [44] given by

$$\tilde{\bar{R}} = \tilde{R} - ((\tilde{R} - \tilde{I} \tilde{p}_i^T) \tilde{a}_i) \tilde{p}_i^T$$  \hspace{1cm} (13)

with $\tilde{a}_i$ the unit normal vector to the affine hyperplane of the ith rule and $\tilde{p}_i$ a point the hyperplane passes through. As $\| \tilde{a}_i \|_2 = 1$, the norm in (12) can be equivalently expressed by

$$\| \sqrt{\tilde{Q}_i (\tilde{R} - \tilde{\bar{R}})} \|^2 = \tilde{a}_i^T (\tilde{R}^T - \tilde{\bar{R}}^T) \tilde{Q}_i (\tilde{R} - \tilde{\bar{R}}) \tilde{a}_i$$  \hspace{1cm} (14)

By mean centering the matrix $\tilde{R}$ with a weighted centroid vector $\tilde{\bar{y}}$:

$$\mu_{\tilde{y}} = \frac{\tilde{R}^T \tilde{Q}_i \tilde{y}}{\tilde{R}^T \tilde{Q}_i \tilde{y}}$$  \hspace{1cm} (15)

i.e., the weighted mean of all samples (in each regressor) with weights given by the membership degrees to the ith rule, obtaining a mean centered matrix $\tilde{R} = \tilde{R} - \mu_{\tilde{y}} \Psi_i$ by columnwise subtraction, the minimization of (14) yields

$$\tilde{a}_i^T (\tilde{R} \tilde{Q}_i \tilde{R}) \tilde{a}_i + \tilde{R}^T \tilde{Q}_i ((\tilde{p}_i - \mu_{\tilde{y}})^T \tilde{a}_i)^2 = \min! \tilde{a}_i, \mu_{\tilde{y}}$$  \hspace{1cm} (16)

where the matrix $\tilde{R} \tilde{Q}_i \tilde{R}$ denotes the Hessian matrix with the inclusion of the target vector $\tilde{y}$ as first column and thus is symmetric and positive definite. Minimization of the first term in (16) is achieved when $\tilde{a}_i$ is associated with the smallest eigenvector of this matrix (as equivalent to the classical matrix eigenvector decomposition problem—see [45]), whereas the minimization of the second term is obviously achieved in the case when $\mu_{\tilde{y}} = \mu_{\tilde{R}}$, as only then this term vanishes.

As $\tilde{a}_i$ is the smallest eigenvector of the covariance matrix and all eigenvectors are orthogonal, it can be realized as the normal vector of the hyperplane solution spanned by the more significant eigenvectors. Thus, the (implicit) normal vector form of the solution plane is given by

$$a_{i0} \tilde{y} + a_{i1} r_1 + a_{i2} r_2 + \ldots + a_{ip} r_p = \tilde{p}_i^T \tilde{a}_i$$  \hspace{1cm} (17)

with $p$ the number of regressors. This is because $\tilde{p}_i$ is a point the hyperplane passes through and thus its scalar product with the normal vector coefficients, i.e., $\tilde{p}_i^T \tilde{a}_i$, yields the right-hand side of the equation. The prediction of the ith local model for a new data sample $\tilde{x}$ is thus given by

$$\tilde{y} = \tilde{p}_i^T \tilde{a}_i - (a_{i1} r_1 + a_{i2} r_2 + \ldots + a_{ip} r_p)$$  \hspace{1cm} (18)

hence the consequent parameter vector $\tilde{w}_i$ of the ith rule is set to

$$\tilde{w}_i = \begin{bmatrix} \tilde{p}_i^T \tilde{a}_i - a_{i1} & -a_{i1} & \ldots & -a_{ip} \\ a_{i0} & a_{i0} & \ldots & a_{i0} \end{bmatrix}$$  \hspace{1cm} (19)

Thus, $\frac{\partial \tilde{E_{i0}}}{\partial a_{i0}}$ belongs to the intercept $w_{i0}$ and the other coefficients to the $p$ regressors.

4. Experimental Section

4.1. Data Collection and Preparation

At the company partner, blast furnace process signals (temperature, pressure, gas composition, and so on) are collected on a Level 1 system and transferred to a relational database via an internal protocol (Level 2). From there, the dataset is further compressed (minute to monthly averages) and stored to tables in an Oracle database (Level 3). There, 5 min averages of the data are
archived for at least half a year. For our investigations, 24 tables were selected to be the most relevant ones. Examples of the available tables are: temperature measurements around the lower blast furnace region, wind temperature, furnace gas composition, physical and chemical data of the charge material, temperature and volume of resulting pig iron and slag, information on maintenance intervals. Data were available for three distinct 6 month periods. The tables were exported to an SQLite format and transferred to the academic partner for further preparation. This preparation included the removal of constant or duplicate columns, detection and interpolation of some outlier intervals probably stemming from sensor errors, and assignment of (later available) material property values to the time the material was processed. As the recording frequency of the process samples was 5 min, the final datasets for the three distinct 6 month periods (used as training, validation, and test data) contained around 45 000 to 50 000 samples each.

The original number of input variables collected and stored (373 in total) was reduced according to an expert-based preselection (see also Section 3.1), in the follow up of several internal discussions about their point of view on the meaningfulness of certain system variables for predicting some of the targets. Therefore, the company experts compiled an excel sheet, in which trivial correlations were marked to be excluded from the investigations. Furthermore, an expert matrix was provided with parameters that were assumed to have the highest impact on the individual target values. This list of parameters was used to develop an ordered subset of input variables. On the one hand, this allows the integration of experts’ knowledge into the modeling procedure. In the case of cooling capacity, the experts also defined an exclusion list, i.e., a list of input variables which should not be used explicitly for modeling purposes (top-down feature selection). In this way, 309 input variables could be excluded, ending up in a set of 64 input variables. On the one hand, this allows the integration of the existing process knowledge, on the other hand, measurements with strong interdependence due to their physical positioning and physical setup were also excluded like, e.g., temperature measurements of the same region.

The maximal past history to be considered in the modeling, i.e., the parameter $K$ in the regression matrix, was defined by experts to be 650 min in the case of hot metal temperature and to be 480 min (i.e., 8 h) in the case of cooling capacity. According to a 5 min resolution of sample recordings, this leads to a time series length of 130 steps in the case of hot metal temperature and silicon concentration and to a length of 96 steps in the case of cooling capacity, making up $130 \times 34 = 4420$ resp. $96 \times 30 = 2880$ columns in the regression matrix $R$. This can be seen as a high-dimensional problem, thus dimension reduction as described in Section 3.2 was an indispensable issue during predictive modeling cycles.

4.2. Evaluation Strategy

The collected dataset from the blast furnace process was divided into three parts following three consecutive years of execution. The data from the first year (2017) containing $\approx 45 000$ samples in 5 min resolution were used as training and validation data of our models: the first half of the data was used for training purposes and the second half of the data was used to elicit the optimal setting of the learning parameters of our modeling method and of NNs models, with which we compared our approach. This half–half split respects the timely order of the data, i.e., the data used for validation always appears timely after the training dataset, which actually mimics the expected prediction behavior for new online samples; this is not the case when using conventional K-fold cross-validation technique[46] as some test folds appear intermediate or even before the folds used for training, and as such do not model real prediction into the future. Data from the second year (2018) were used as a separate test dataset to elicit how the model (built on 2017 data) is expected to behave in the future. Data from the third year (2019) were used to check the further behavior of the models (further into the future) and especially whether there is an increase in the error to be expected over a longer time (due to some implicit, hidden dynamics, changes in the system relations etc.). In this sense, the 2019 dataset can be seen as a (pessimistic) long-term test case regarding the expected robustness of the models over time.

Regarding the validation stage in our evaluation, we conducted a full grid search over the two essential learning parameters in our modeling procedure: one is the factor $f_{\text{tie}}$ in the statistical threshold of the rule evolution criterion, which steers the number of rules generated; it has been varied from 0.5 to 4.0 in steps of 0.35, as also recommended and successfully used for various application scenarios in the study by Lughofer et al.[20]—please note that there is no further parameter for the consequent learning phase needed, according to the parameter-free WTLS approach. The second parameter is the number of input variables to be chosen from the ranked list $L_{\text{out}}$, as outputted by Algorithm 1. As mentioned at the end of Section 3.2, we iterated over the full list $L_{\text{out}}$, starting with one regressor (the most important one) and successively adding regressors one-by-one and perform modeling steps on the selected ordered subset. In this sense, we obtained a 2D parameter grid as exemplarily shown in Figure 5, where the rows denote the input dimensionality and the columns...
The nature of the models not allowing interpretability. In this sense, our approach can be seen as superior due to offering of interpretable rules, see earlier discussion. Thus, we could check whether a black box model is the real observed values. This leads to the values as shown exemplarily in Figure 5. For a final model selection, we applied a punished minimum error criterion respecting model complexity, where higher complex models are punished as they are more likely to overfit on new future data and furthermore have less interpretability for experts. That is, we selected that model $\hat{y}_{ij}$ from the grid which leads to the minimal value of $\text{MAE}(\text{pun})_{ij}$ given by

$$
\text{MAE}(\text{pun})_{ij} = e^{0.05(\#\text{Inputs} + \#\text{Rules})} \cdot \text{MAE}_{ij}
$$

The effect of the punished minimum error criterion becomes apparent when inspecting the upper two marked grid points: in the case when selecting the model with minimal MAE, the model with 3 inputs and 9 rules with an MAE of 0.45 would be selected; however, when applying the punished criterion, the model with a slightly higher MAE of 0.5, but with a lower complexity would be selected, as having just two inputs and five rules.

To compare our modeling procedure for hot metal temperature and silicon concentration, we applied a feedforward NN approach (implemented in MATLAB’s Deep Learning toolbox), for which we varied the number of hidden layers from 1 to 4 in steps of 1, the number of neurons from 1 to 50 in steps of 5 and also the input dimensionality according to the ranked list of inputs produced by Algorithm 1 (so, the same ranked list as used for our modeling procedure). This leads to a 3D parameter grid with the same final model selection as done for our models, see earlier discussion. Thus, we could check whether a black box model with more complexity in terms of multilayer structure and higher nonlinearity degree (as achieved by the NNs) may actually lead to better accuracy and does actually pay off—for paying off, the model error should be significantly lower because of the black box nature of the models not allowing interpretability.

In addition, we compared our modeling results for the cooling capacity with models obtained by the following modeling techniques:

1) Classical multivariate linear regression models (linear regression) with embedded Tikonov regularization for improved robustness in the case of noise; 2) Fuzzy systems training with conventional axis-parallel rules [47] (fuzzy axis-parallel (classical)) to see the effect of rule rotation; we varied the vigilance parameter (steering the number of rules) from 0.1 to 0.9 in steps of 0.1; 3) Local linear model trees (LoLiMoTs) after the hierarchical split-based construction, as discussed in the study by Nelles [48]; we varied the maximal number of splits from 1 to 15 in steps of 1; 4) Feedforward NNs with sigmoid activation functions (Feedforward NNs), where we varied the number of neurons from 5 to 100 in steps of 5 and the number of layers from 1 to 10 in steps of 1; hence, they also cover a deep-layered functionality [49]; 5) Ensembles of regression trees after the “fitensemble” implementation in MATLAB (FitReEnsembles): Bayesian hyperparameter optimization was used to optimize several learning parameters internally, such as the method-type switching between “bagging” and “boosting,” the number of learning cycles, the minimal leaf size and the learning rate; note that in case of bagging, the resulting models are the well known and widely used random forests [30]; 8) Bagged fuzzy systems (Bagged Fuzzy): we also generated a whole bag of fuzzy systems with our procedure on different bootstrap replications; to see whether they can increase model accuracy of single fuzzy systems due to bootstrapping-based reduction of noise; the number of bags were varied between 2 and 30 in steps of 1; 9) Gradient-boosted fuzzy systems (Gradient-Boosted Fuzzy); we also generated fuzzy systems based on successive residuals to explain the remaining (unexplained) content of the target after the idea of gradient boosting [52]; we varied the number of boosted models between 2 and 10 in steps of 1; 10) Extreme learning machines (ELMs) using a flat layer structure, but with a large-span number of neurons [53]; we used a robust regularization method embedded [54] as implemented in MATLAB’s toolbox; we varied the number of neurons from 5 to 1000 in steps of 5.

5. Results

5.1. Hot Metal Temperature and Silicon Concentration

Figure 6 shows the model errors on the separate test dataset from the year 2018 in terms of mean absolute errors (y-axis) over the different predictions horizons (x-axis), left for the silicon concentration, right for the hot metal temperature.

The prediction horizons ranged from 80 to 140 min, as this turned out to be the most interesting horizons for the expert. The expected values of these targets for the next tapping should be seen in advance to react properly in time when the values do not meet the expectations of the operators working with the condition monitoring system. Typically, the next racking is expected in about 100 min, thus we indicated this prediction horizon as the most interesting one with a vertical marker in the plots. As shown in the figure, in the case of the silicon concentration, the three modeling techniques performed similarly, especially for the 100 min prediction horizon, whereas in the case of hot metal temperature, the classical fuzzy modeling with axis parallel rules ended up with significantly worse errors than our approach for most of the horizons. Also the NN models were worse for most of the horizons.

All in all, it is remarkable that NNs cannot really outperform the two fuzzy variants despite allowing higher model complexity while neglecting any interpretability. In this case, our approach can be seen as superior due to offering of interpretable rules, see the following paragraphs. Another remarkable issue is that the increase in model errors from an 80 min to a 140 min prediction horizon is just about 10% for both targets, and nearly negligible (≈1%) in a horizon range of 100–140 min.
A more detailed insight into the predictive performance of our methods was achieved by looking at the predicted versus observed target value trend lines over time, especially for visualizing how well the principal trends of the tappings can be predicted therein. In the observed trend line, each tapping can be realized by a step-change followed by a constant phase. Figure 7 shows a snapshot of the trend line over a time frame of about two and a half weeks (i.e., for 25 000 min corresponding to 5000 samples) during year 2018 as produced by our modeling procedure for the 100 min prediction horizon, left for silicon concentration, right for hot metal temperature.

Obviously, several principal up and down trends can be well predicted by our method, in total leading to a correlation coefficient of about 0.75 for hot metal temperature and of about 0.72 for silicon concentration. In this context, it was interesting for the company experts to see how many of the real trends of the tappings (over the complete test dataset) could be actually predicted correctly: for instance, if the target value from one to the following time step increases, then our model should also ideally predict an increased target value, at least after a reasonable small delay; this would then count as a correct prediction subject to a delay. Evaluating this aspect it finally turned out that in the case of the silicon concentration 74.75% of the trends could be correctly predicted, and this with an average delay of 19.34 min, whereas for the hot metal temperature 80.36% could be correctly predicted with an average delay of 22.78 min. Note that, in the case of NNs, only 70.63% and 75.89% of the trends with similar delays could be correctly predicted, respectively. It is remarkable that, when calculating the error solely on the correctly recognized trends, we ended up with an MAE of 5.12 vs 9.89 on all samples) for hot metal temperature and with an MAE of 0.053 mass% (vs 0.098 mass% on all samples) for silicon concentration—this means that, once the model recognizes the correct trend, a much lower (halved) error of the prediction can be expected.

Recall that the primary motivation for our modeling procedure was to have a final (implementable) model which also offers interpretable components and parameters, where a specific focus was put on the influences of certain variables on the model.

Figure 6. Model error trends over various prediction horizons (x-axis), left for silicon concentration, right for hot metal temperature; the results for our modeling procedure (solid line) are displayed, in comparison with the two best related state-of-the-art methods (conventional fuzzy systems with axis-parallel rules and NNs).

Figure 7. Observed versus predicted target trend lines produced by our modeling procedure over a time frame of about 17 days for silicon concentration (left) and for hot metal temperature (right).
outputs in different modes/ regimes. Hence, we analyzed the final fuzzy model for hot metal temperature prediction in detail. Thereby, only five input variables turned out to be sufficient for a best possible prediction of hot metal temperature in 100 min (with 9.89° and 5.12° error on correct trends, respectively, outperforming state-of-the-art works in related literature). These five most influential inputs have been KL_P.WF.GES at the current point of time, OD.TEUF at the current point of time, KL_P.RAST at the current point of time, and KL.P_RAST 225 min earlier (see Table 1 for the physical meanings); furthermore, the hot metal temperature at the current point of time turned out to be also an important input. The fuzzy partitioning of these inputs into LOW and HIGH regions is shown in Figure 8, where we omit to show KL.P.RAST 225 min earlier.

Figure 8. Fuzzy partitioning of the four input variables for a 100 min ahead prediction model of hot metal temperature.

| Rule #1: | Rule #2: |
|---|---|
| **IF** RE.RE_TMP(t-0) is LOW AND OD.TEUF(t-0) is LOW AND KL.P.RAST(t-0/225) is HIGH **THEN** RE.RE_TMP(t+100) = | **IF** RE.RE_TMP(t-0) is HIGH AND OD.TEUF(t-0) is LOW AND KL.P.RAST(t-0/225) is LOW **THEN** RE.RE_TMP(t+100) = |
| +0.6941 * RE_RE_TMP(t-0) | +0.5048 * RE_RE_TMP(t-0) |
| +0.02685 * KL_P.WF.GES(t-0) | +0.02121 * KL_P.WF.GES(t-0) |
| -0.055 * OD.TEUF(t-0) | -13.36 * OD.TEUF(t-0) |
| -0.01405 * KL_P.RAST(t-0) | -0.0173 * KL_P.RAST(t-0) |
| +0.01125 * KL_P.RAST(t-225) | +0.01705 * KL_P.RAST(t-225) |
| +409.9 | +696.8 |

| Influence | Influence |
|---|---|
| 1017.62 | 740.08 |

| Rule #3: | Rule #4: |
|---|---|
| **IF** RE.RE_TMP(t-0) is LOW AND OD.TEUF(t-0) is LOW AND KL.P.RAST(t-0/225) is LOW **THEN** RE.RE_TMP(t+100) = | **IF** RE.RE_TMP(t-0) is LOW AND OD.TEUF(t-0) is LOW AND KL.P.RAST(t-0/225) is LOW **THEN** RE.RE_TMP(t+100) = |
| +0.7441 * RE.RE_TMP(t-0) | +0.4698 * RE.RE_TMP(t-0) |
| +0.02653 * KL_P.WF.GES(t-0) | +0.01889 * KL_P.WF.GES(t-0) |
| -5.882 * OD.TEUF(t-0) | -1.015 * OD.TEUF(t-0) |
| -0.02365 * KL_P.RAST(t-0) | -0.02757 * KL_P.RAST(t-0) |
| +0.0184 * KL_P.RAST(t-225) | +0.0198 * KL_P.RAST(t-225) |
| +341.5 | +756.9 |

| Influence | Influence |
|---|---|
| 1090.9 | 688.8 |

| Rule #4: | Rule #4: |
|---|---|
| **IF** RE.RE_TMP(t-0) is LOW AND OD.TEUF(t-0) is LOW AND KL.P.RAST(t-0/225) is LOW **THEN** RE.RE_TMP(t+100) = | **IF** RE.RE_TMP(t-0) is LOW AND OD.TEUF(t-0) is LOW AND KL.P.RAST(t-0/225) is LOW **THEN** RE.RE_TMP(t+100) = |
| +0.4698 * RE.RE_TMP(t-0) | +0.4698 * RE.RE_TMP(t-0) |
| +0.01889 * KL_P.WF.GES(t-0) | +0.01889 * KL_P.WF.GES(t-0) |
| -1.015 * OD.TEUF(t-0) | -1.015 * OD.TEUF(t-0) |
| -0.02757 * KL_P.RAST(t-0) | -0.02757 * KL_P.RAST(t-0) |
| +0.0198 * KL_P.RAST(t-225) | +0.0198 * KL_P.RAST(t-225) |
| +756.9 | +756.9 |

| Influence | Influence |
|---|---|
| 688.8 | 40.70 |

Figure 9. Four fuzzy rules extracted from data for the predictive modeling of the hot metal temperature termed as “RE.RE_TMP.”
because it showed the same partitioning as the current KL_P_RAST.

It is also noteworthy that for KL_P_WF_GES no partitioning took place (which was in-line with the experts' expectations), thus it can always be seen as a “Do Not Care” part in the rule antecedents and hence is omitted when showing the rules below. Four rules have been automatically extracted from the data, and are shown in Figure 9.

The IF part in the rules denotes the preconditions, which define a certain local region in the feature space; for instance, Rule #1 represents a local model in the case when the current hot metal temperature (RE_RE_TMP) is HIGH, OD_TEUFE is LOW etc.; in this region, the actual model is described by the consequent part (after the THEN statement), which is a linear local regression fit according to (6). In this fit, the parameters are not directly comparable, because we are acting on original input variable values, thus their medians and ranges are different (hence, they operate on a different scale)—this is possible, because our modeling method is fully scale-invariant, thus data need not be normalized in advance. Thus, we listed the actual local influence of each variable with respect to each rule in a separate column. It can be observed that the current hot metal temperature has the highest influence on the hot metal temperature in 100 min, followed by KL_P_WF_GES and KL_P_RAST in both time lags, whereas the influence of OD_TEUFE is comparably negligible. In this sense, our approach can also act as a local feature reduction approach, ending up with a final compact, interpretable model largely depending on just three inputs, the current hot metal temperature, and the inputs KL_P_WF_GES and KL_P_RAST. The last two process variables refer to the cooling capacity of the tuyres and the cooling capacity of the bosh area. From a physical point of view, it seems to be possible that heat transfer processes in those parts of the furnace influence the hot metal temperature (with further evaluations to be undertaken). Furthermore, the importance of the autoregressive aspect were also expected by the experts.

Finally, we used an additional dataset from year 2019 to investigate how much loss in model accuracy can be expected when
letting the model make predictions in a more distant, 2 years ahead time frame without any recalibration phases (note that the models were built on data from the first half of year 2017). The model errors over the various prediction horizons are visualized in the right plots of Figure 10, upper for silicon concentration, lower for hot metal temperature, which can be directly compared with the left plots showing the errors on the year 2018 data.

Obviously, the increase in error from year 2018 to year 2019 is negligible as around 0.005–0.01 mass% in the case of silicon concentration, and about 0.2–0.4 °C in the case of hot metal temperature for our modeling procedure. Hence, the intrinsic dependencies between system variables do not change over time. It is also notable that our approach shows the most stable results for hot metal temperature, outperforming the other two methods for all prediction horizons.

5.2. Cooling Capacity

For this target, a sort of explanation model should be established which explains the current cooling capacity based on values of a subset of system variables (preselected by experts) occurring up to 8 h before. The integration of self-recursivity, i.e., of past, lagged target values for own prediction, was not allowed, as it would have led to trivial correlation models and thus no helpful explanation. Figure 11 shows the results in terms of mean absolute errors (in percent according to the target range) as well as correlation coefficients between observed and predicted target values for the cooling capacity modeling problem. The left four columns in this table show the results based on 30 variables preselected by experts, the right four columns show the results based on all variables minus explicitly excluded ones. These were not allowed to be considered as they also would have led to trivial correlations, mainly because of redundant measurement channels in the dataset.

The rows in this table are the different methods that have been tested, as explained in the itemization of Section 4.2. Obviously, our fuzzy modeling methods perform better than related state-of-the-art works for both criteria, correlation coefficient and mean absolute error, thus highlighted in bold font, with the exception on test data from 2019, where LoLiMoT can perform slightly better in the case of using all minus excluded variables. The bagging as well as gradient boosting strategies for our models performed very similarly to the single fuzzy model case, thus could not significantly improve their performance. Hence, they can be neglected in further considerations, because they are hardly interpretable ensemble models (with more than 10 fuzzy models as members). It is also interesting to see that the variables selected by experts are not performing much worse than when using all variables minus the obvious redundant ones, losing about 0.03–0.04 in terms of correlation intensity and about 1–2% in terms of errors. This confirms the long-term expertise of the company experts, which variables may best affect the cooling capacity. Finally, it should be noted that the errors on both, the year 2018 and year 2019 dataset could be significantly reduced by an automated model off-set correction procedure which adds a constant bias term to all of its model predictions (to compensate a possible process shift). For our fuzzy model, this led to the final errors of 7.84% on the year 2018 data set and of 8.00% on the year 2019 data set, which were approved as a good performance by the company experts.

The observed versus predicted target values for our (offset-corrected) model over the complete duration of the year 2018 dataset is shown in Figure 12.

It can be seen that there are several phases in which the cooling capacity goes down toward 1000 kW, as highlighted by some ellipsoids: in most of these phases, the prediction also showed a significantly decreasing trend, hence indicating a reliable explanation model for the target.

The fuzzy partition input structure of the latter is shown in Figure 13, left for the input OD_FLAMTMP(t-25), the flame temperature value 25 min earlier, and the input RE_ANA_SI(t-150), the silicon concentration 150 min earlier; for both, three linguistic terms LOW, MEDIUM, and HIGH were sufficient to describe the local regions of the input space. These were the only two inputs needed in the final selected model. For the experts, the importance of RE_ANA_SI for explaining the cooling capacity was a bit surprising at first glance, but after a deeper consideration turned out to be a valuable new insight.

| Method                | Expert, Non-Recursive Test Data 2018 | Expert, Non-Recursive Test Data 2019 | All minus excluded, Non-Recursive Test Data 2018 | All minus excluded, Non-Recursive Test Data 2019 |
|-----------------------|-------------------------------------|-------------------------------------|-----------------------------------------------|-----------------------------------------------|
|                       | Corr. | MAE % | Corr. | MAE % | Corr. | MAE % | Corr. | MAE % | Corr. | MAE % |
| Linear Regression     | 0.674 | 0.138 | 0.494 | 0.124 | 0.731 | 0.130 | 0.558 | 0.111 |
| Fuzzy (Ours)          | 0.712 | 0.127 | 0.546 | 0.119 | 0.746 | 0.126 | 0.572 | 0.110 |
| LoLiMoT               | 0.693 | 0.144 | 0.485 | 0.128 | 0.740 | 0.126 | 0.571 | 0.109 |
| Feedforward NNs (deep)| 0.692 | 0.144 | 0.485 | 0.128 | 0.725 | 0.129 | 0.563 | 0.113 |
| FitREEnsembles       | 0.599 | 0.144 | 0.275 | 0.129 | 0.601 | 0.130 | 0.419 | 0.123 |
| Bagged Fuzzy          | 0.713 | 0.132 | 0.546 | 0.122 | 0.746 | 0.127 | 0.571 | 0.110 |
| Gradient Boosted Fuzzy| 0.712 | 0.127 | 0.543 | 0.119 | 0.742 | 0.126 | 0.566 | 0.110 |
| SparseFIS             | 0.700 | 0.137 | 0.519 | 0.122 | 0.730 | 0.127 | 0.565 | 0.111 |
| ELMs                  | 0.649 | 0.149 | 0.472 | 0.133 | 0.746 | 0.128 | 0.572 | 0.113 |

Figure 11. Model errors and correlations between observed and predicted targets of various modeling methods for predicting/explaining the cooling capacity; the left four columns are based on expert-selected variables, the right four columns are based on “all minus explicitly excluded” variables.
Figure 12. Observed versus predicted target value trend lines produced by our modeling procedure over a time frame of the whole year 2018 for the cooling capacity.

Figure 13. Fuzzy partitioning of the two input variables (left flame temperature, right silicon concentration) of an explanation model for cooling capacity.
Four rules were required to explain the cooling capacity; these are shown in Figure 14, together with the local influences of the two input variables in the regions represented by the four rules.

It is interesting to see that, whenever the flame temperature is HIGH (Rules 1 and 3), it also has a high influence on the cooling capacity (much higher than silicon concentration), whereas when it is getting to MEDIUM (Rule 4), its influence becomes equal to silicon concentration and when it is LOW (Rule 2), its influence becomes negligible (compared with the influence of silicon concentration). This monotonic trend with opposite sign for silicon concentration was seen as plausible by the experts. A concrete current value of the cooling capacity can be thus explained through these influences: for instance, if the flame temperature 25 min before was LOW, the reason for the current cooling capacity value is solely because of a (high/low) value of silicon concentration 150 min earlier.

6. Conclusion

We proposed a data-driven modeling approach for an automated extraction of prediction and explanation models for hot metal temperature, silicon concentration, and cooling capacity in an ironmaking blast furnace process, based on a larger variety of internal process variables together with their past history, leading to time-series based forecasting problems. Due to the combination of a process variable ranking method to reduce the curse of dimensionality effect and the specific type of fuzzy systems as model architecture with universal approximation capabilities, it was possible to achieve a satisfactory predictive quality over a longer future time frame and a compact linguistic interpretability of the internal model components (in form of IF–THEN rules) at the same time. It was even possible to outperform the related state-of-the-art methods such as deep NNs, ELMs, or ensembled regression trees.

Acknowledgements

This work was supported by the COMET-K2 Center “Symbiotic Mechatronics” of the Linz Center of Mechatronics (LCM) funded by the Austrian federal government and the federal state of Upper Austria.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

Research data are not shared.

Keywords

blast furnaces, explainability, interpretability, prediction models, steel production

Received: February 2, 2021
Revised: April 9, 2021
Published online:

[1] S. Pal, A. Patra, P. R. Padhee, Process Modeling for Steel Industry, I K International Publishing House, Hoboken, NJ 2018.
[2] S. Amano, T. Takarabe, T. Nakamori, H. Oda, M. Taira, S. Watanabe, T. Seki, ISIJ Int. 1990, 30, 105.
