Real-Time Defect Prevention in Continuous Casting

Authors

P. Hänisch, SMS digital GmbH, Düsseldorf, Germany

B. Dutta, SMS group Inc., Pittsburgh, Pa., USA

R. Murthy, SMS digital GmbH, Düsseldorf, Germany

C. Kirmse, SMS digital GmbH, Düsseldorf, Germany
christoph.kirmse@sms-digital.com

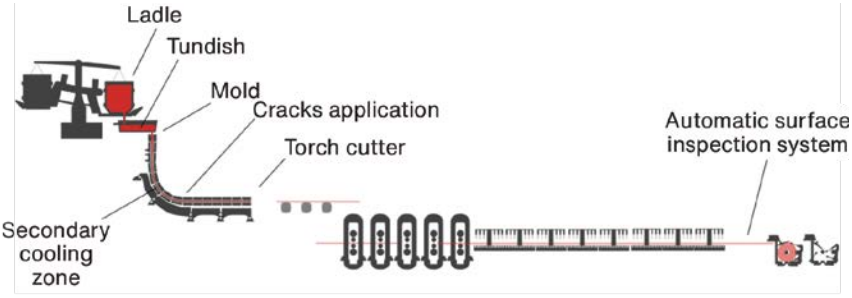
Digital technologies are transforming industry at all levels. Steel has the opportunity to lead all heavy industries as an early adopter of specific digital technologies to improve our sustainability and competitiveness. This column is part of AIST's strategy to become the epicenter for steel's digital transformation, by providing a variety of platforms to showcase and disseminate Industry 4.0 knowledge specific for steel manufacturing, from big-picture concepts to specific processes.

The continuous casting process is the most widespread form of producing steel. In recent years, over 96% of steel has been produced with the continuous casting process.[1] A wide range of products can be produced with this process, such as thin slabs, thick slabs or billets, to name a few. A schematic overview of the continuous casting process is shown in Fig. 1. The liquid metal from the ladle is poured into the tundish and flows from there into the mold. The mold is water-cooled, and during the downward movement the liquid metal solidifies. This process starts from the outside of the strand. Through further cooling in the secondary cooling zone, all the liquid metal becomes solid, so that it is completely solidified at the torch cutter. During this process between the mold and the torch cutter, surface defects can occur. These defects are detrimental to the quality of the steel product. Poor-quality slabs need to be post-processed (e.g., by grinding or scarfing). The post-processing adds extra costs and reduces the weight of the final product. If the quality of a slab is too poor, it must be scrapped.

To predict and prevent defects from occurring, the Cracks Preventer application was developed. The Cracks Preventer predicts surface defects before they occur. It uses signals from the mold, the secondary cooling zone and L2 databases to predict the defects. By suggesting appropriate countermeasures, defects can be avoided. This supports the operators in producing high-quality products and leads to less post-processing and less scrapping of slabs and thus lower production costs and less CO_2 emissions.

Figure 1

Production line from caster to coil. The slabs could be stored between casting and rolling.



Literature Review

Classical Approaches for Defect Detection in Continuous Casting

Casting defects such as longitudinal cracks can reduce the quality of a slab or lead to strand breakouts. To detect such casting defects, much effort has been made in previous works to find solutions which can detect and prevent those defects. The following section gives an overview about the state of the art regarding defect detection.

In the work of Liu et al.,[2] a temperature field was developed based on the measured temperature signals from the temperature monitoring system. The visualization of the temperature field by the temperature thermograph is shown in Fig. 2.

The temperature fields result from interpolation between the individual thermocouples. The temperature thermograph makes it possible to visually detect a longitudinal crack by determining the temperature differences between thermocouples. For this purpose, a current temperature difference is calculated using Eq. 1:

$$D_{!}[x, y] = T_{!}[x, y] - T_{"}[x, y] \quad \text{(Eq. 1)}$$

If abnormalities occur, such as a longitudinal crack, the thermograph shows a window with negative temperatures. Fig. 3 illustrates this with an example.

With the help of a computer image processing algorithm, the formation and movement of a longitudinal crack is recorded in the thermograph. For this purpose, abnormal regions are separated from normal regions using a segmentation algorithm. Based on the segmented regions, properties of the longitudinal crack are extracted and combined with the properties of a stationary production process (e.g., constant cast speed) and properties of a non-stationary production process (such as cast speed lowering). It turns out that longitudinal cracks have their own specific pattern (Fig. 3), which means that this type of defect can be detected with relative certainty.[2]

Another classic method for detecting longitudinal cracks is the use of principal component analysis according to Lieftucht et al.[3] Here, the measured temperatures from the mold monitoring system, which are determined, for example, by thermocouples, are used. It is assumed

Figure 2

Visualization of temperature field for mold copper plates.[2]

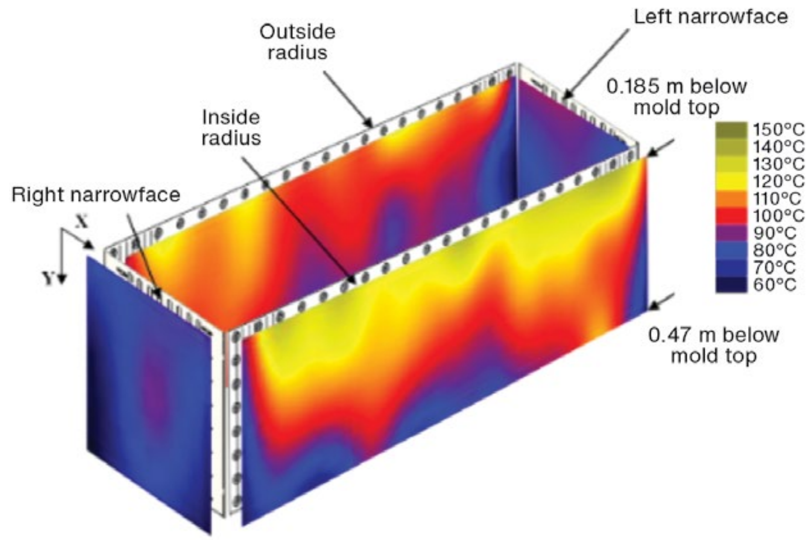


Figure 3

Temperature drop in the thermograph when a longitudinal crack occurs.[2]

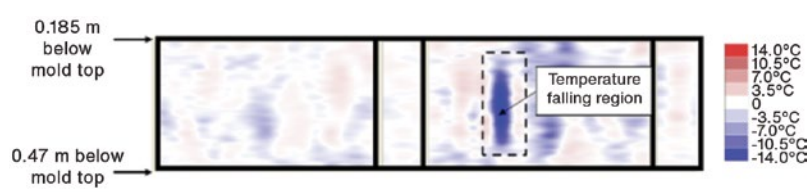
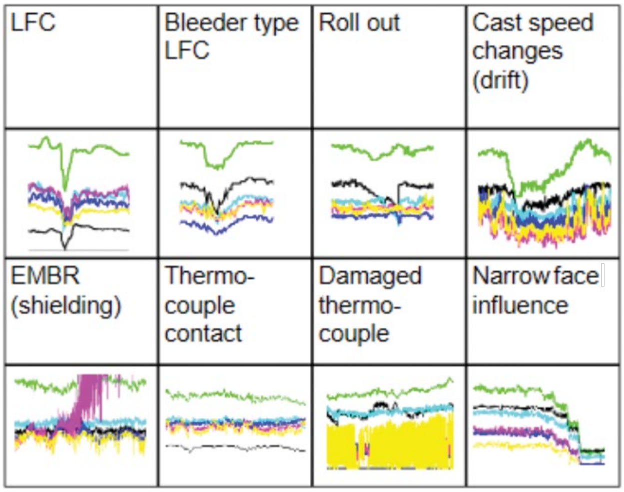


Figure 4

Different types of thermocouple signal patterns.[3]



that longitudinal cracks are expressed by patterns in the temperatures, which have the shape shown in Fig. 4.

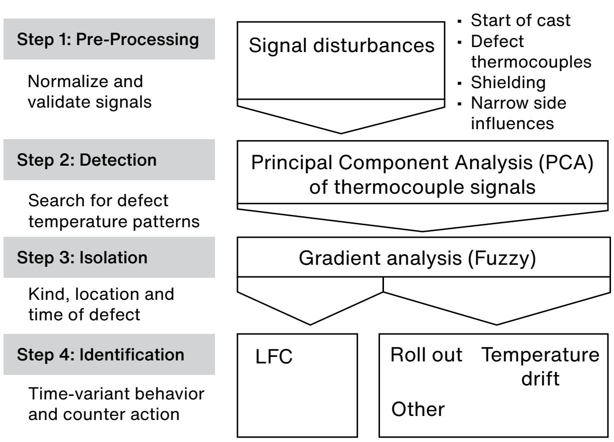
Fig. 5 shows the flow chart of the model for longitudinal crack detection using principal component analysis. The first step is filtering the signals. In this context, only stationary process states are selected. This means that states such as casting start and casting end are neglected and not subjected to the main component analysis. This step erases erroneous measurement results that are caused, for example, by defective thermocouples.

In a second step, principal component analysis is applied to the training data, which consists of normalized thermocouple signals. The result of this main component analysis are uncorrelated eigenvectors, which represent the variance in the training data set. By applying principal component analysis to new data, a comparison of the variances is possible. The comparison enables predictions as to whether a longitudinal crack will occur or not. In the third and fourth step, a gradient analysis of the thermocouple temperatures is used to determine the type of abnormality and its location.[3]

The approaches mentioned above only consider one data source at a time. The first approach took temperature distributions into account, while the second approach took thermocouple temperatures into account. However, there are a variety of reasons for the formation of casting defects. This requires an approach that can consider different data sources such as time series data for sensor data and scalar data for chemical compositions or steel grades. The approaches described above neither take into account the various data sources, nor is it possible to determine the causes.

Figure 5

Hybrid, multi-step model structure for LFC defect detection.[3]



Methodology

Product Architecture

The Cracks Preventer consists of several components (see Fig. 6):

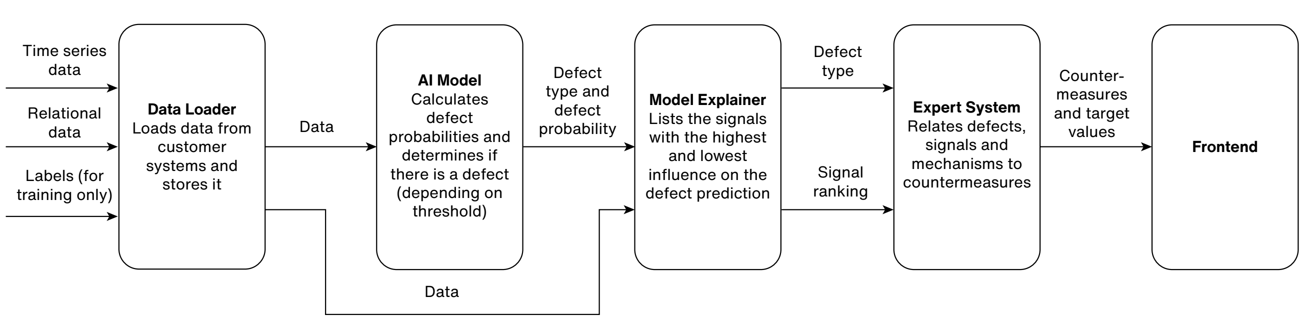
- Data Loader.
- AI Model.
- Model Explainer.
- Expert System.
- Frontend.

It combines automation systems (Data Loader) with AI Models (defect detection, model explainer) and Expert Systems (countermeasures). The flow of the application is the following: Data from the customer's automation system (e.g., level 1 or level 2) is loaded into the Data Loader, where it is stored. The Data Loader provides the data to the training and the production environments.

The next stage in the Cracks Preventer application is the AI Model. It is trained with data from the Data Loader that also includes the labels from the surface inspection system. In the present case, that surface inspection system is a processing step later, but it is also the first time most defects from continuous casting can actually be detected. In production, data is fed from the

Figure 6

Product architecture.



Data Loader to the AI Model without the labels. The data that includes time series and relational data is preprocessed to map the time series data to the relational data. The AI Model predicts the probability of a defect on a particular section of the strand. Additionally, for each slab, a prediction is performed to evaluate whether the slab needs post-processing or not. Strand sections or slabs are marked as having a defect if the defect probability surpasses a pre-defined threshold.

Every time a defect is detected, the defect type and probability along with the corresponding signal data is sent to the Model Explainer. The Model Explainer identifies for every defect the signals that have the highest influence on the defect probability. Thus, a signal ranking is created for every single defect that the model predicts.

This signal ranking along with the defect type is sent to the Expert System, where metallurgical knowledge is stored. It derives the mechanism that led to that particular defect from the defect type and the influencing signals. Knowing the mechanisms that led to the defect, suitable countermeasures can be suggested.

The results (defect type, signal ranking, values of the signals, countermeasures, etc.) are sent to the application's Frontend, where they are displayed. Caster operators can view the Frontend on a screen in the pulpit. The result values can also be fed to the customer's automation system to automatically trigger the countermeasures. Additionally, configurable reports can be downloaded from the Frontend.

The Cracks Preventer can be deployed in the cloud or on premises. The Frontend is usually shown in the pulpit on a screen. The only requirement is a connection to the back-end systems and a screen where the Frontend can be shown in standard browser software.

The AI model ingests different types of data. These include:

- Time series data: Signals from mold and secondary cooling zone like mold level, temperatures, or cast speed. The signals can be of different length and frequency.
- Relational data: Chemical analysis from the ladle furnace or degasser and steel grades.
- Labels: From customer's defect classification system. The labels are only required for training; they are not available in production.

The signals stem from different locations in the plant (ladle, mold, secondary cooling zone, surface inspection system, etc.), see Fig. 1. A defect can only be detected by the surface inspection system on the rolled coil at the end of the process. The genealogy is important as the position of a defect that is detected on the coil needs to be related to its corresponding position on the slab and the time stamps when this position has been in the mold and the secondary cooling zone. Through rolling, cutting and other processing steps, this can be quite challenging. In the mold and secondary cooling zone, the signals usually have a time stamp. However, the position on the strand with the defect passes first through the mold and then through the secondary cooling zone. The time difference between these two positions needs to be considered by using the casting speed so that only the signals are used that relate to the defective position on the strand.

The information from the surface inspection system might be available only weeks or months after the slabs have been produced as the slabs might be stored and only later rolled to coils. The Cracks Preventer can predict a number of different surface defects such as holes or cracks. Any defect that is detected by the surface inspection system can be predicted by the Cracks Preventer. The prediction is executed at two different times during production. The first prediction is performed anywhere between the mold and the secondary cooling zone area, depending on the defect and its occurrence region. Each defect has a different area of origin. This prediction covers a pre-defined section of the strand and predicts any defects occurring on this section. It is done periodically with a custom interval. It ensures that defects are captured early in the process and that appropriate countermeasures can be taken. These countermeasures can have an influence on the subsequent sections of the strand (for any countermeasures taken in the mold or secondary cooling zone) or on the following heats (for countermeasures taken in the ladle furnace or degasser). The second prediction is executed after a slab has been cut. It predicts whether there is a defect on any part of the entire slab. This information can then be used to determine whether the slab needs to be sent for further treatment like scarfing or grinding before it can be rolled.

Deep Learning Model: The signals that are used for the prediction of defects are time series signals and non-time series signals. Therefore, a mixed model is used that can utilize both data types. Examples of such models are long short-term memory (LSTM) networks, gated recurrent units (GRUs) or convolutional neural networks (CNNs). They belong to the class of deep learning models. Deep learning models usually have a large number of layers, which enables them to learn complex relationships between the input data and the outputs.

LSTMs and GRUs belong to the class of recurrent neural networks (RNNs). They are specifically designed for problems where the current input relies on the previous one. LSTMs have, compared to other networks, longer training times and require more memory for training. GRUs train faster than LSTMs as they have fewer internal gates. This has the drawback that they do not perform well on tasks that require long-distance relations. Both the LSTMs and the GRUs are hard to interpret, as they have internal states. The CNNs are easier to interpret and faster to train than RNNs. From the three model types tested on the existing data set, they worked best in terms of model performance. Therefore, the CNN was selected for defect prediction.

The CNNs consist of multiple layers. Each layer has several hyperparameters. The number of layers as well as the layers' hyperparameters can be adjusted to change the model architecture. The layers can be regarded as bricks that can be arranged individually. Bricks can be added to the model (e.g., a new layer for time series signals) or removed from it to change the model architecture. Based on metallurgical knowledge, signals can easily be added or removed from the model. This also allows for short experimentation cycles without the necessity to change large parts of the training environment.

The time series data will be split into time windows with a fixed length. The time window length can vary for different signals. For the training data, the time windows for the signals can overlap, as shown in Fig. 7. Thus, more training data can be generated. The sampling frequency can also vary for different signals. While the mold level might be sampled with a high frequency, the cast speed could be sampled with a lower frequency (or vice versa). The length of the time windows are an important parameter for the accuracy of the prediction and one that will be tuned during the model training. If the windows are too small, the predictive areas of the signals are missing. If the windows are too large, there is too much noise in the signals.

The data is split into three sets: training, dev and test sets. The training set is used to fit the model weights and parameters. During training, the dev set is used to optimize these weights and parameters. After the training has finished, the test set is used to check how well the model can generalize to unseen data.

For time series signals, the order of the input signals matter and they cannot be shuffled randomly. Especially, data from one heat cannot be present in two different sets. This would overestimate the performance of the model greatly as the algorithm would memorize each heat.

The output of the Machine Learning model is a probability value between 0 and 1 that describes how likely it is that a defect occurs. This probability needs to be translated into a flag that determines whether a defect is predicted. Thus, a threshold value is set. If the defect probability is above the threshold value, then a defect is predicted. For probabilities below the threshold value, no defect is predicted. The threshold value is model specific.

Figure 7

Time windows for three different time series signals.

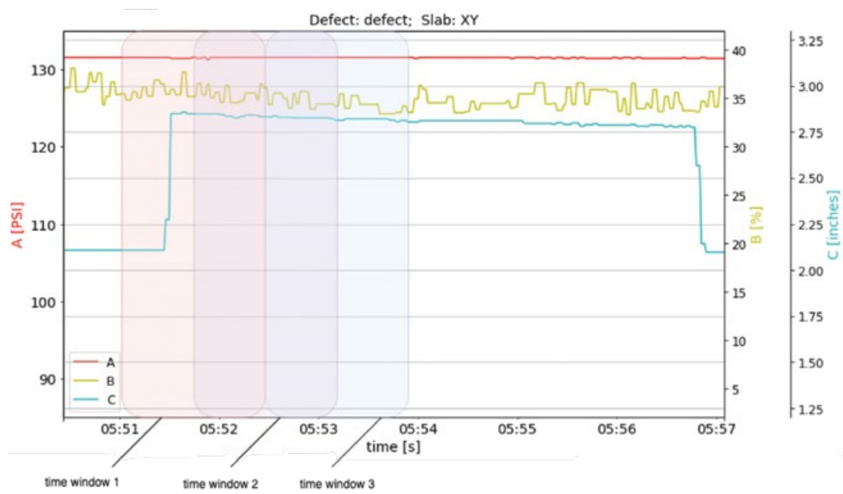


Figure 8

Basic concept of model explainer for one signal (modified from Reference 4).

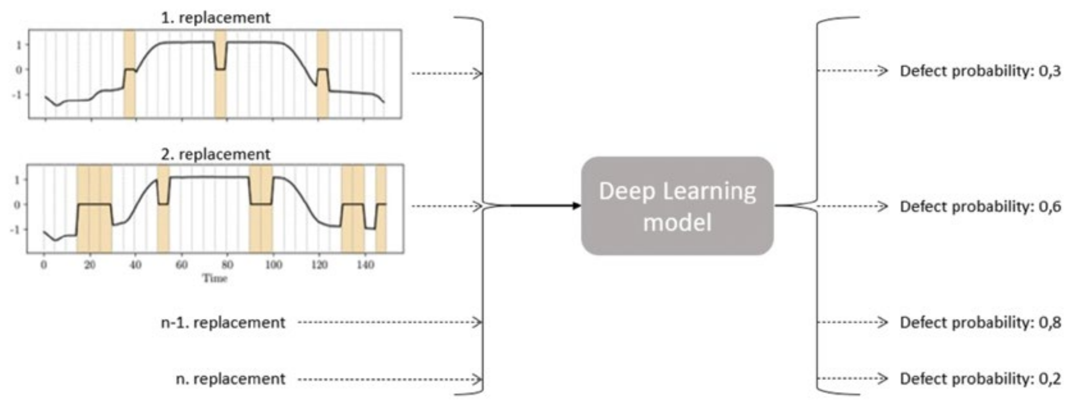


Figure 9

Impact of considering sample weights in a linear model.

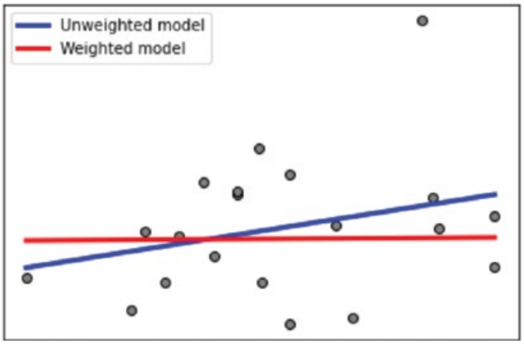
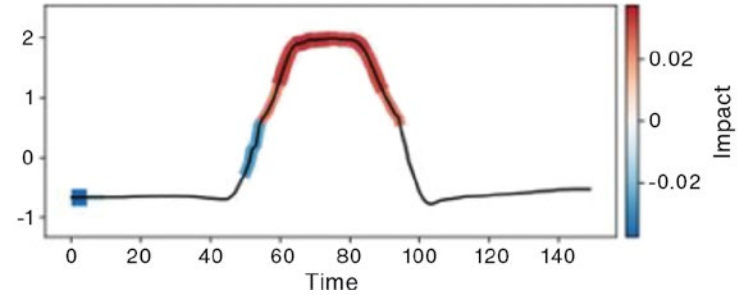


Figure 10

Importance of each time series segment marked as red and blue areas.[5]



The Machine Learning models are trained on historical data for which defect labels are available. Classical performance metrics for classification problems can be used to determine the quality of the models. These include confusion matrix, accuracy, precision, recall and F1 score. Custom performance metrics can also be implemented.

Model Explainer: It is important to not only perform a real-time prediction of the casting defects, but to also find the most influencing factors for a particular prediction to get the root causes of the predicted defect. On top of the root causes, there can be attached mechanisms that explain the metallurgical origin of a defect as well as countermeasures. Those countermeasures can be applied to the casting process by the operators to avoid defects from occurring in the future.

To get the most influencing factors of a prediction from the Machine Learning model mentioned in the section above, a Model Explainer has been implemented, motivated by Guidotti et al.[4] The idea behind the Model Explainer is to make the deep learning model result explainable; one could say to get rid of a black box model. The basic concept behind the explainer is to replace segments of a time series signal and make a prediction after a segment has been replaced. Fig. 8 shows this schematically.

There are different strategies that can be applied to generate the above-mentioned segment replacements. These are local mean, global mean or 0 replacement.[4]

For each replacement, there is a defect probability and a weight calculated. In the developed approach, different strategies have been considered to calculate the weights, such as:

- Uniform approach.
- Exponential kernel weight with l2 distance.
- SHapley Additive exPlanations (shap) kernel weights.

After calculating predictions and weights for a single time series with *n* repetitions, the indexes of the original segments and the predictions are fit to a linear model where the calculated weights are considered as sample weights. The impact of considering the calculated weights as sample weights is shown in Fig. 9.

Following that, the coefficient from the fitted linear model is extracted and used to calculate the segment importance. The result of this process can be seen in Fig. 10. Red areas show important areas for a prediction for a time series segment. On the other hand, blue areas show less important segments of a time series signal.

The concept described so far considers one signal. Usually, data sets with more than 100 time series signals per batch are fed into the model. The next step is to get the most important time series signals. Therefore, the described procedure is repeated for all signals in a data batch. Fig. 11 shows this schematically for the time series signals A, B and C.

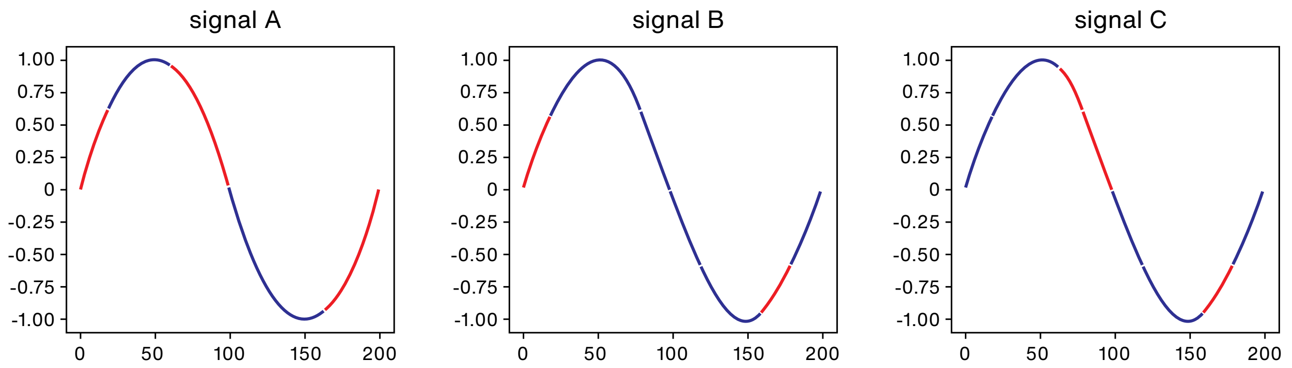
According to the developed Model Explainer, those segments are important which are marked in red. In the end when the calculation has taken place for all signals, a ranking will be calculated.

Results

As discussed in the previous section, the Machine Learning model returns a defect probability that is used to determine whether a particular defect occurred in the analyzed section. This result is then sent to the Frontend, which is shown in Fig. 12. On the top left, the last analyzed slab is shown in which a defect detection has occurred. It contains the defect type that has been detected and additional information about the slab on which the defect has been detected. This includes the heat ID, the slab ID and the steel grade. Below the last analyzed slab is the 72-hour history section. It contains all slabs that had a defect during the last 72 hours (this time period can be adjusted). Here the same information is displayed as in the last analyzed slab section. On

Figure 11

Segment marking for three time series signals.



the right-hand side of the defect history is the 72-hour Slab Statistics section. In this section, the total number of produced slabs and the number of defected slabs are shown. Additionally, for each defect type the number and percentage of defected slabs is displayed. This helps in identifying whether particular defects occurred more often than others.

The rest of the screen contains detailed information about the most recently detected defect. On the top right, the signal(s) with the highest influence on the defect are shown. These signals were identified by the Model Explainer. The operators can inspect the signal values and discover any unusual behavior of the signals. Below the root cause signals the corresponding mechanisms are displayed. The mechanisms were identified in the Expert System, which contains metallurgical know-how about each defect type. Next to the mechanism, the countermeasures are shown. They give valuable recommendations on how to change process settings in order to avoid the defect in the future. A connection to the automation system can also be established for an automatic execution of the countermeasures.

Figure 12

Frontend screen. On the top left the last slab with a defect is shown. Below the detailed 72-hour history for all slabs with defect prediction is displayed. On the right, the most influential signals for the current defect are shown. The mechanisms and countermeasures of the current defect are listed below that.

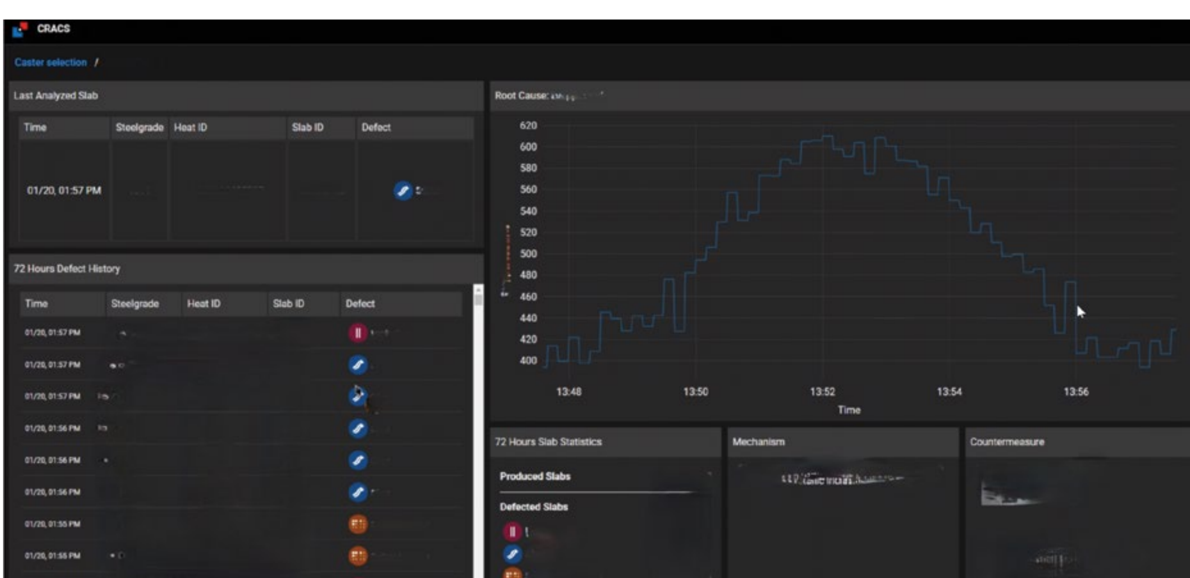
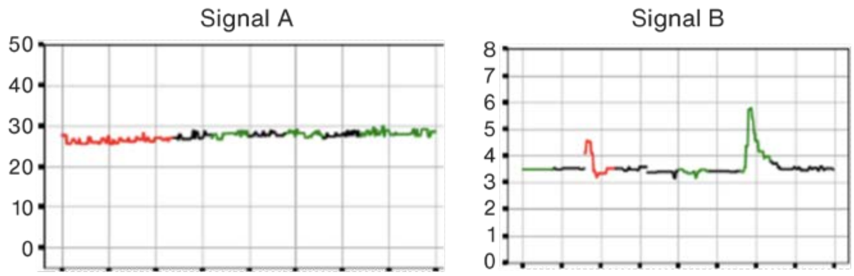


Figure 13

Marking of time series segments by the model explainer. The red segments have a high influence on defect prediction. The green segments have a low influence on the defect prediction.



The Model Explainer results for two signals (A and B) are shown in Fig. 13. The Model Explainer marks the relevant sections of the time series signals according to their importance for the defect prediction. Sections that support the defect prediction are marked in red and sections that oppose the defect predictions are marked in green. As can be seen in Fig. 13 for signal A, the beginning of the time window is marked as having a high influence on the defect prediction. Segments in the middle and end have a low influence on the defect prediction. There are also segments that have neither a low nor a high importance on the defect prediction. Those segments are marked in black. It is interesting to observe that for signals B the first peak supports the defect prediction, while the second peak opposes it.

The signals are ranked according to their influence on the defect prediction. This signal ranking together with the signal values are then shown in the Frontend and sent to the Expert System to identify mechanisms and countermeasures. The marked segments and the signal ranking can be analyzed by quality engineers to adjust the casting process.

Conclusions

During the continuous casting process of steel, surface defects can occur. The Cracks Preventer application is a combination of Machine Learning models and an Expert System to predict and prevent those defects. The Machine Learning model uses chemical data of the heat, steel grade data, as well as time series signals from the mold and secondary cooling zone. The model predicts whether a defect occurs on a particular section of the strand or a slab. When a defect has been predicted, the Model Explainer finds the most influential signals for the prediction. This information is sent to the Expert System to find the underlying mechanisms and suggest countermeasures to avoid the defect from occurring in the future.

The results of the Machine Learning model and the Expert System are sent to a web application where it can be accessed by the operators in the casting pulpit and by quality engineers to analyze the predicted defects and their root causes.

The Machine Learning model can be extended to predict more surface defects. Additional signals from the casting process or any upstream process step can be included in the model to increase the prediction accuracy.

References

1. "2022 World Steel in Figures," World Steel Association, 2022.
2. Y. Liu, X. Wang, Y. Sun, F. Du, Y. Gao, F. Wang and J. Wang, "A Visual Detection Method of Longitudinal Crack Based on Computer Image Processing," *Archives of Metallurgy and Materials,* Vol. 63, No. 2, 2018, pp. 673–680.
3. D. Lieftucht, M. Arzberger, M. Reifferscheid and J. Schlüter, "Online Prediction of Longitudinal Facial Cracks in Thin Slab Casting Using Principal Component Analysis," *Journal of Iron and Steel Research International Supplement 1,* 2008, pp. 255–259.
4. R. Guidotti, A. Monreale, F. Spinnato, D. Pedreschi and F. Giannotti, "Explaining Any Time Series Classifier," *2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI),* Atlanta, Ga., USA, 2020.
5. F. Mujkanovic, "Explaining the Predictions of Any Time Series Classifier," bachelor of science thesis, Hasso Plattner Institut, 2019.

This paper was presented at AISTech 2023 — The Iron & Steel Technology Conference and Exposition, Detroit, Mich., USA, and published in the AISTech 2023 Conference Proceedings.