Explainable Steel Quality Prediction System Based on Gradient Boosting Decision Trees

JANNE TAKALO-MATTILA\textsuperscript{1}, MIKKO HEISKANEN\textsuperscript{1}, VESA KYYLÖNEN\textsuperscript{1}, LEENA MÄÄTTÄ\textsuperscript{2} AND AGNE BOGDANOFF\textsuperscript{2}

\textsuperscript{1}VTT Technical Research Centre of Finland, 90570 Oulu, Finland
\textsuperscript{2}SSAB Europe Oy, 92100 Raahe, Finland

Corresponding author: Janne Takalo-Mattila (janne.takalo-mattila@vtt.fi)

This work was supported in part by the Model-Based Optimization for Efficient Use of Resources and Energy (MORSE) Project, and in part by the European Union’s Horizon 2020 Research and Innovation Program under Grant 768652.

**ABSTRACT** The steelmaking industry is one of the most energy-intensive industries and is responsible for 4% of the world’s total greenhouse gas emissions. Solutions to improve operational efficiency can therefore bring major improvements to the overall environmental performance of the entire industry. This article proposes a novel steel quality prediction system based on gradient boosting trees that can be used to predict the quality of steel products during manufacturing. The prediction system enables the detection of possible surface defects in the early phase of the manufacturing process, thus avoiding costly and time-consuming manufacturing efforts to address defective products. In this study, we trained a prediction model with data collected from an SSAB Europe steelmaking plant in Raahe, Finland. From the 296 process parameters measured in the liquid steel stage of steelmaking, we selected 89 input features to train and test the prediction model. The model was then integrated into a quality monitoring tool (QMT) to utilize real-time manufacturing data in its predictions. The validation process showed that the prediction model can find more than 50% of defective steel products by marking only about 10% of the steel products as potentially at risk of surface defects in plate rolling. This can potentially save time in the quality control phase and improve process efficiency. To gain more insights into the model predictions, we used SHAP (SHapley Additive exPlanations) to find a potential connection between the process input parameters and surface defects.

**INDEX TERMS** Explainable artificial intelligence, machine learning, quality prediction, steel manufacturing.

**I. INTRODUCTION**

Iron and steelmaking is considered one of the most energy-intensive industries in the world, contributing to about 5% of the world’s total energy consumption and accounting for 4% of the world’s total greenhouse gas emissions [1]. However, there are various approaches to improve the industry’s operational energy efficiency [2] and reduce CO\textsubscript{2} emissions, such as using direct hydrogen reduction instead of coke-based blast furnaces [3].

Another possible way to reduce emissions and energy consumption in the iron- and steelmaking industry is to improve operational efficiency by enhancing product quality and avoiding defective products. If problems in a steel product are detected at an earlier stage, corrective actions can be taken and additional problems avoided later on [4]. Often, the inspection of steel products is done by human operators, which can lead to inaccuracies due to different inspection criteria [5]. The automatic detection of slab defects with machine vision has also been proposed as a solution [6]. Steel production quality can be controlled as well by keeping process parameters and chemical concentrations within desired limits. Indeed, a rule-based expert system that controls the operation of blast-furnace can contain up to 1200 rules [7]. These rules are typically easy to interpret and therefore commonly used in various industries where trust and decision transparency are essential. However, there may be hundreds of process parameters and measurements in the steelmaking process that affect final product quality, so the process yield may remain unclear even for domain experts. Machine learning methods provide a solution to construct experience from complex data, gain new information from

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.
process measurements, and make predictions about new data samples. It is often used in complex multivariate problems for anomaly detection, classification, and prediction and is therefore suitable technology for the steelmaking industry, where large amounts of measurements are made, often at high intervals. Just to mention a few examples, Ko et al. [8] used machine learning to detect anomalies in machinery engines by combining engine inspection data with after-sales service data. Caggiano et al. [9] used deep learning to classify defected items in selective laser melting process.

One of the key challenges in utilizing machine learning models in the steelmaking industry is their lack of transparency. Current machine learning models are typically effective in making predictions, but they are black-boxes in nature [10]. Recently, various techniques to explain the decisions made by machine learning have been developed [11], [12]. Explainability in particular provides both a way to audit the decisions made in tightly regulated industries and make decisions that users can trust. In [13] Nunes et al. concluded that two most important purposes for explainability is to provide transparency i.e. explain why model ended up with the decision and secondly demonstrate effectiveness of the model.

In this paper, we propose a novel approach to predicting the surface quality of steel products using a gradient boosting–based machine learning algorithm. The proposed machine learning model uses real production data from the liquid steel phase of the steelmaking process and combines process data with manually inspected product surface quality data from two different quality control points. By integrating the prediction model into quality monitoring tools, process operators can receive assistance in selecting which steel products need to be inspected more carefully after the continuous casting phase and whether corrective actions, such as slab scarfing, should take place. In addition, the quality prediction model can guide operators in fine-tuning process parameters by providing insights into the most influential parameters that have the greatest effect on product quality. Reliably predicting production quality can generate economic benefits by helping manufacturers avoid the costly and time-consuming correction of defective products or by implementing corrective measures earlier in production.

The rest of this paper is organized as follows. Section 2 presents earlier works related to our study, focusing on recent advancements in machine learning–assisted steelmaking. In this section, we also present our own advancements in the field. Section 3 describes the steelmaking process used in this study, the type of data, and the setting in which the quality prediction model was used. This section also presents the details of the gradient boosting–based steel quality prediction model itself. The implementation and integration of the quality prediction model is discussed in section 4. Section 5 presents the model’s evaluation and prediction performance using different datasets. Lastly, section 6 concludes the paper.

II. RELATED WORKS

This research reports on the development of methods for reliable and explainable steelmaking property prediction and classification. There are various solutions for the automatic prediction of steelmaking properties and the detection of defects in steelmaking products. This section introduces the
literature on machine learning–based methods for steelmaking prediction. In the context of this literature analysis, the term *property prediction* may include binary classification (defective or normal products) for slabs or other steelmaking products, multi-category property classification, or regression-based methods that predict one or more numerical features.

Continuous casting is an important part of the steelmaking process, with 99% of U.S. steel production alone using this technology [14]. Continuous casting process has significant effect of the overall quality of the steel products [15]. In [16], the quality of continuous casting slabs was predicted using a time series classification. Mold level fluctuations were measured at 0.5-s intervals, and the data were combined with inspection machine data. The predictions used both fully convolutional networks (FCN) and long-term short-term memory (LSTM). Thakkar et al. [17] likewise predicted longitudinal facial cracks in slabs that occur in the continuous casting phase. They tested various machine learning algorithms, such as decision trees, random forest, gradient boosting, and a support vector machine (SVM). Centerline segregation is another, internal defect in steel slabs that can occur in continuous casting. In [18], centerline segregation was predicted using SVM and particle swarm optimization (PSO). The researchers presented the significance of physicochemical parameters using the weights of the fitter PSO-SVM model and found the average casting speed to be the most influential parameter in predicting centerline segregation. Lieber et al. [19] predicted the final quality of steel bars using self-organizing maps (SOM) and a naïve Bayesian classifier. In [20], machine learning using quantile hyper-spheres (QH-ML) was proposed to classify steel surface defects.

In addition to predicting defects in the steelmaking process, machine learning methods can also be used to predict the microstructure and mechanical properties of produced steel. For instance, Xie et al. [21] predicted the mechanical properties of steel based on the elemental compositions and process parameters of the entire hot-rolling process. According to their work, deep neural networks are the most accurate means of predicting product properties. Local interpretable model-agnostic explanations (LIME) have also been used to interpret predictions. Ensemble machine learning models can be used as well to predict the properties of steel products as presented in [22], where algorithms that rely on boosting strategies, such as gradient boosting decision trees, LightGBM, and XGBoost, were found to perform better than traditional single-regression methods, such as linear regression, Ridge regression, and Lasso regression. In addition, Zhang et al. [23] demonstrated that LightGBM provides both best predictive performance, but is also highly computation-ally effective compared to random forest, deep feedforward neural network and SVM. Gue et al. [24] also used machine learning to predict several properties of steel products, such as yield strength, plasticity, and tensile strength. Their approach used 27 input features, including both chemical compositions and process parameters. The prediction model used ordinary least square (OLS), SVM, regression trees, and random forest methods.

Improving the quality of steel products and avoiding defects clearly reduces waste in the production process. In addition, predictions on steel yield can also be made in the steelmaking process. Yield determines the percentage of metal, scrap, and iron ore converted into steel slabs. Laha et al. [25] used support vector regression (SVR) to predict steel yield, reporting that prediction precision can meet steel production requirements.

Our work continues the above research results and focuses on the prediction of surface defects based on machine learning with a gradient boosting algorithm. The main contributions of our work are as follows:

- **Novel gradient boosting–based surface quality prediction model:** In [22], ensemble methods showed promising results for prediction accuracy; therefore, we use a gradient boosting decision tree–based machine learning model for the prediction of product surface quality early in the steel manufacturing process. We utilize the production, parameter, and surface quality inspection data of an actual steel plant as our dataset.
- **Transparency in model decisions:** Improved explainability of the classification model using SHAP (SHapley Additive exPlanations) and analysis of the process features which will have the biggest effect on final product quality.
- **Model deployment to a real production environment:** We have deployed the model in a real steelmaking environment and integrated it with a quality monitoring system.
tool (QMT) to be used by process engineers [26]. The model can use close to real-time process measurements from the plant to make predictions. Local explanations of the model decisions can also be integrated into the QMT tool, providing insights to further improve the steelmaking process.

- Comprehensive evaluation of the model’s performance with actual production data: We also evaluated the model with a dataset from the actual steelmaking process and comprehensively analyzed its prediction performance across different settings and datasets.

III. STEEL PRODUCT QUALITY PREDICTIONS

This section briefly describes the steelmaking process and the methods currently used to monitor slab surface quality in the presented steelmaking facility. We also present the datasets used in this study and offer detailed descriptions of the data preprocessing phases. Finally, we show how the architecture and hyperparameters of our gradient boosting decision tree algorithm were designed and how the model was trained for steel product classification.

This study utilizes data from a steel production facility located in Raahe, Finland. The Raahe steel production facility is part of SSAB Europe and has a crude steel production capacity of 2.8 Mtons. The total number of employees on site is 2,500 [27].

A. PROCESS OVERVIEW

The data used in this study was collected from the steelmaking process shown in figure 1. This figure is a simplified version of the actual production process, showing only the most relevant parts for our experiments. The process begins at the blast furnace, which is a continuously operating shaft furnace. A mixture of coke, iron ore, and fluxes are fed into the top of the blast furnace and combined with an oxygen flow. The purpose of the blast furnace in the process is the reduction of iron ore in the hot metal [28]. Sulfur is also removed in a desulphurization process before the material moves to the basic oxygen processing (BOF) converter [29]. In the BOF converter, oxygen is blown through the top lance and inert gas is purged from the bottom to refine the hot metal into crude steel. A ladle furnace follows the BOF converter as an intermediate steel processing unit. Its role is to further refine the steel’s chemistry and temperature. The process also uses vacuum degassing, where excess hydrogen, nitrogen, and carbon are removed in a vacuum [30]. Alternatively, at this state, Composition Adjustment by Sealed argon bubbling – Oxygen Blowing (CAS-OB) [31] can be used to homogenize and control the steel’s composition and temperature [32].

After this processing of the molten metal, steel slabs are produced in a continuous casting process. The slabs are inspected after casting for the first time and then sent for plate rolling. After the plate rolling phase, surface defects are inspected in the quality inspection gate for a second time. This is the point at which the operator performs the final manual inspection of the steel end products (i.e., plates) to ensure that they meet the desired quality requirements.

B. DATASETS FROM THE STEEL MANUFACTURING PROCESS AND QUALITY INSPECTION

The datasets used in this study were collected from the process described in section 3A in two separate batches. The first dataset is historical data collected from the steelmaking plant from May 2017 to November 2019. This dataset contains a total of 225,461 rows without any preprocessing. The second batch of data were collected during the period when the model was tested in a production environment. The second batch of data were collected during the period when the model was tested in a production environment. The testing period began in December 2019 and ended in May 2020.

Figure 2 presents the datasets collected from the steelmaking process and their dates. The model’s training was done with Data_1, and it was tested with Data_2 that has been collected between December 2019 and May 2020.

The datasets were collected from four separate subsets, with each storing measurements from different parts of the steelmaking process. The subsets were:

- Steel tundish data: 4 features
- Superheat data: 5 features
- Steel segment data: 16 features
- General liquid steel process data before plate rolling: 251 features

We did not use data from the plate rolling phase for this study. There are two quality inspection sites in the steelmaking process. The first inspection is performed after the continuous casting phase, when a process operator inspects the surface of the casted slab. The second quality inspection takes place after the plate rolling, as shown in figure 1 where the operator inspects the plate surface for possible defects. Three main surface defects exist, namely longitudinal, sliver, and transversal defects. In this study, we were interested in

| TABLE 1. Data batches from the steelmaking process. |
| Dataset | Period | Role in the study |
|---------|--------|------------------|
| Data_1  | 5/2017–11/2019 | Training, validation, and test data in model development |
| Data_2  | 12/2019–5/2020 | Test data |

FIGURE 3. Dataset division in validation.
classifying steel products into normal and defect categories and therefore grouped all defect types into one category for binary classification.

The produced slabs are identified by heat number (HEAT), which is automatically incremented in the process. During each heat, multiple slabs are produced. Slab number (MOTHER SLAB) identifies a specific slab during each heat. Thus, individual slabs and plates created from a mother slab are uniquely identified by (HEAT, MOTHER SLAB) identifiers. Some process measurements are made several times during slab production. In these cases, we aggregated the measurements and created new features, including mean, maximum, and minimum values of the corresponding production measurements. Finally, after preprocessing, the production data contains a total of 296 features, both numerical and categorical.

The datasets were divided into several subsets to train and evaluate the machine learning model. Table 2 shows the data subsets separated from data batches Data_1 and Data_2. In this study, we were interested in the performance of the classification model with datasets containing all slabs produced in the time period. We were also interested in whether the prediction model’s performance would deteriorate over time. As such, we used Data_1 for model training, validation, and testing. Data_2 was only used to test the classification model’s performance with completely unprecedented data from a different production batch.

### C. GRADIENT BOOSTING DECISION TREE–BASED QUALITY PREDICTION

The goal of our quality prediction model was to classify steel products into normal and defective products based on process data. The model utilizes process data collected before the plate rolling stage, meaning it can be used, for instance, after continuous casting as a support tool to decide if a slab can be sent to the plate rolling phase without additional processing (e.g., scarfing).

The task of the quality classification model is to decide class membership $y$ based on corresponding process data $x'. The prediction model was trained with dataset $X = (x_1, y_1), \ldots, (x_n, y_n)$, where $x_i, y_i$ is a known example pair in dataset $X$. The model predicts single output as follows:

$$\hat{y} = f(x)$$

where $\hat{y}$ is the predicted target for known parameters $x$ using the learned function $f$, $x_i$ is an $m$-dimensional vector whose components are called features or sets of input variables; it is mapped with target variable $y_i$.

In this study, we used gradient boosting decision tree algorithms to develop the steel quality prediction model, as implemented with the LightGBM framework [33]. As mentioned in the section II, algorithm has shown promising results in various domains compared to similar methods. This framework supports various gradient boosting algorithms, including the traditional gradient boosting decision tree (GBDT), dropouts meet multiple additive regression trees (DART), and gradient-based one-side-sampling (GOSS). In this work, all the supported gradient boosting algorithms are compared as a part of the hyperparameter search process.

Gradient boosting algorithms belong to a group of ensemble methods in which multiple decision trees are combined to make a decision, typically in a regression or classification task. By combining the decisions made by multiple decisions trees, the overall accuracy should on average be better than any individual member of the ensemble [34]. Ensemble learning model $\varphi$ uses the aggregation function $G$ to aggregate learned functions $f_1, f_2, \ldots, f_n$ to predict single output, as follows:

$$\hat{y} = \varphi((x)) = G(f_1, f_2, \ldots, f_n).$$

In this study, we specifically used gradient boosting ensemble methods. In gradient boosting decision trees, new weak models are added to the ensemble sequentially and trained with the residual of the whole ensemble. A gradient descent algorithm is then used to reduce the error residual at each boosting step. In addition to a traditional gradient boosting algorithm, this study tested DART and GOSS [35] algorithms.

To train and test the gradient boosting–based quality classification model, dataset $D_{11}$ was divided into two separate sections, $D_{11, \text{train}}$ and $D_{11, \text{test}}$. The first 80% of this dataset was reserved for training and the last 20% for testing the model with unseen data. Dataset $D_{21}$ was used only for the model’s testing. A test set is always held out for the final model evaluation with completely unseen data to gain a realistic performance estimate of real operational use. After the dataset’s separation, we used $K$-fold cross-validation to divide the training set into $k$ groups of samples. The model was trained with the $k - 1$ fold, and one group was used for its testing. In particular, we used time series $K$-fold cross validation, where the first $k$ folds were used to train the model and $k + 1$ fold was used as a validation set. The used cross-validation dataset division is presented in figure 3.

Finding the type of machine learning model that provides optimal prediction performance is a challenging and time-consuming task. We thus used a grid search to find an optimal hyperparameters from specified set of possible parameters. Hyperparameters are variables that define the structure of a model and how the model is trained. The grid search builds
a new model for each parameter combination and evaluates it with a validation set. Grid search tries every hyperparameter combination from table 3 and tests the performance of the model using cross-validation with selected performance metrics. Therefore, grid search is typically a computationally demanding way to find optimal parameters. We selected the parameters that provided the best performance in the cross-validation and tested the resulting models with the test parts of datasets $D_{11}$ and $D_{21}$. Table 3 shows the list of parameters that we tested in the model training and parameter tuning. As a scoring metric in the hyperparameter search and model validation, we used the area under the curve–receiver operating characteristics (AUC–ROC) curve. ROC graphs are one means of analyzing classifier performance. In an ROC graph, true positive rate $tpr$ is plotted on the y-axis between 0.0 and 1.0, and false positive rate $fpr$ is plotted on the x-axis between 0.0 and 1.0, leaving $tpr$ estimated as

$$tpr = \frac{\text{defects correctly classified}}{\text{Total number of defects}},$$

and $fpr$ as

$$fpr = \frac{\text{Normals incorrectly classified}}{\text{Total number of normals}}.$$

The graph visualized $tpr$ and $fpr$ at decision threshold $\tau$ and thus visualized a trade-off between $tpr$ and $fpr$. The binary classification model predicted the probability of negative and positive classes, where the sum of the probabilities was always 1. The probability was then converted into class labels based on $\tau$, where prediction values exceeding the threshold were converted to positive classes.

A perfect classifier would have $tpr$ 1.0 and $fpr$ 0.0, resulting in an AUC of 1.0. AUC can be used to compare the performance of multiple classifiers with a single value. A coin toss binary classifier would have an AUC of 0.5 [36].

Based on the grid-based hyperparameter search, table 4 presents the best hyperparameters for the classification model, with AUC as a validation metric.

### TABLE 3. Model hyperparameters for grid search.

| Hyperparameter            | Values                  |
|---------------------------|-------------------------|
| Estimators                | 500                     |
| Learning rate             | 0.001, 0.01, 0.05, 0.1  |
| Subsample                 | 0.6, 0.7, 0.8, 0.9      |
| Minimum child samples     | 2, 3, 4, 5, 6           |
| Scale positive weight     | 1, 2, 4, 6, 8, (normal slabs/defective slabs) |
| Subsample frequency       | 2, 5, 10, 20            |
| Colsample by tree         | 0.5, 0.6, 0.7, 0.8, 0.9 |
| Number of leaves          | 31, 300, 900, 1200      |
| Max depth                 | -1.5, 63                |
| Max bin number            | 10, 100, 255            |
| Feature fraction          | 0.5, 1                  |
| Lambda L1                 | 0, 0.5, 1               |
| Boosting                  | GBDT, DART, GOSS        |

### D. DECISION EXPLAINABILITY AND FEATURE SELECTION

In addition to accurately predicting steel slab quality at the earliest possible phase in the manufacturing process, one of this study’s main objectives was to improve the transparency of how the model uses its input features to make predictions and why individual steel products receive specific predictions. There are two separate aspects in the model’s explainability, namely local and global explanations.
Local explanation for individual steel product quality prediction allows process operators to understand whether something can be changed in the process to improve quality and which features are the main reasons for the predicted quality. Global explanations summarize the effect of input features for model predictions, so it was particularly useful in this model’s development.

In this study, SHAP [11] was used to provide explanations for the model’s decisions. SHAP is a model-agnostic interpretation method, which means that it can treat machine learning models as a black-box, thus providing an opportunity to replace the existing model if needed without compromising its ability to explain decisions. SHAP uses Shapley values [37] to calculate the contribution of each individual feature to the prediction. In our case, Shapley values described how each feature contributes to the risk of a defective steel product. The ability to provide explanations for each sample was a major advantage of SHAP in our study, as operators could utilize the model to understand which parts of the production process have the greatest impact on quality when problems arise.

We also utilized SHAP in the feature selection process. The original dataset $D$ contains process data samples $x'$, which form an $m$-dimensional vector that without feature elimination $m$ is 296. The model was first trained with dataset $D_{11\text{-train}}$ using the cross-validation process presented in section 3C, and 100 of the most important features were selected based on their Shapley value, with any highly correlated features removed. To determine correlation, one feature was selected and paired with another feature, and if their correlation was greater than 0.85, the other feature was removed. After the feature selection and removal, 89 features were left in the dataset.

### IV. IMPLEMENTATION AND INTEGRATION WITH A QUALITY MONITORING TOOL

We implemented the steel quality prediction model and data processing procedure with Python 3.7.3. We also applied the gradient boosting–based model from section 3C with the LightGBM-framework [38]. The model explainability described in section 3.D was implemented with the SHAP TreeExplainer library [39]. In the model development phase, the data were fetched from various Excel files, which were originally exported from the steel manufacturing plant’s production databases.

The model we developed was deployed to the steel manufacturing plant, where process engineers could make use of its prediction capabilities. We deployed the model using a QMT developed in our earlier work [26], which allows the deployment of prediction models in production and process development.

With the QMT tool, information from manufacturing processes can be preprocessed and visualized to an end user, such as a process operator, process engineer, or business manager. Using the QMT tool, different types of statistical modeling methods can easily be used in real industrial applications, especially in product quality monitoring.

The QMT tool consists of two separate parts. First, the QMT server part allows data access to external databases and integration into calculation models. Currently, the QMT supports models implemented in Python, C++, or R and rules or equations implemented in the Mathematical Expression Toolkit Library (ExprTk) [40]. The second part of the QMT tool is a web-based user interface that presents quality information in colors, with red indicating a malfunction in the process, yellow warning of possible malfunctions, and green indicating normal process operations. In the main view, the tool shows overall product quality, and users can navigate to the process phases and models to assess reasons for malfunctions.

In this study, we integrated the quality prediction model presented in section 3C into the QMT tool. The QMT server used the prediction model, implemented as a Python module, that calculated the probability of defective steel products. Real-time measurements $\hat{x}$ from the liquid steel phase were used as model input to determine model predictions $\hat{y}$. We replaced the rule-based color coding with decision threshold–based color coding. Specifically, if decision threshold $\tau$ was above the predefined limit, steel product in the QMT was colored red. Steel items close to the threshold were shown in yellow and items far from the threshold in green. Figure 4 presents an example user interface, with the probabilities of steel product defects displayed to the user with the above color coding. Multiple slabs are produced during one heat, so one row in the user interface contains multiple colors. When the user selects a specific slab, the most important features related to the predicted quality are also listed. The list of the most important features is ordered based on Shapley values.

### V. RESULTS AND DISCUSSION

The steel quality prediction model should provide insight into potential problems in steel production and help operators identify products that may or will have surface defects during the remaining production phases. To this end, our model’s evaluation needed to demonstrate how it would work

---

**TABLE 4. Best hyperparameters for the model.**

| Hyperparameter          | Value |
|------------------------|-------|
| Estimators             | 500   |
| Learning rate          | 0.001 |
| Subsample              | 0.7   |
| Minimum child samples  | 6     |
| Scale positive weight  | 4     |
| Subsample frequency    | 5     |
| Colsample by tree      | 0.9   |
| Number of leaves       | 31    |
| Max depth              | 63    |
| Max bin number         | 100   |
| Feature fraction       | 0.5   |
| Lambda L1              | 1     |
| Boosting               | GDBT  |
in different production settings. Therefore, we used two different datasets (presented in Table 2) relevant to production and that did not participate in the model training. To create the most realistic evaluation environment possible, we also evaluated the performance of the prediction model with data (Data_2 in Table 2) from a completely different dataset.

In this study, there was a large imbalance between the negative (normal steel products) and positive (defective steel product) classes, which is common in many real-life binary classification applications. Our datasets also contained far more normal than defective steel product samples. One way to mitigate this problem was to give more weight to the positive samples using the "Scale positive weight" parameter in the LightGBM framework.

Imbalanced datasets also present challenges in selecting the appropriate performance metrics to analyze and compare classification models. The ratio between the number of correctly classified samples and the total number of samples is called accuracy, and many researchers often consider it the most reasonable performance metric. However, for an imbalanced dataset, this metric can be extremely misleading and show overly optimistic results [41]. Therefore, in this study, we used AUC-ROC metrics to test performance. The AUC-ROC curve measured the prediction performance of the model with all classification thresholds. As a single scalar metric to represent the prediction model’s performance at a select threshold, we also used the Matthews Correlation Coefficient (MCC). The MCC shows the correlation between true and predicted values and is always between $-1$ and 1, with 1 representing the perfect classifier, 0 a random classifier, and $-1$ a classifier with inverse predictions. The MCC uses all cells (true positive ($tp$), true negative ($tn$), false positive ($fp$), false negative ($fn$)) from the confusion matrix with a select decision threshold and is calculated as follows:

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}}$$ (5)

Efforts to avoid sample misclassification always depend on the application [42]. In this study’s context, if the model incorrectly classifies a normal steel product as a defective product ($fp$), the time that a human operator manually spends analyzing that slab can be counted as cost ($fp_{cost}$). These costs $fn_{cost}$ stem from an extra plate rolling phase or sending a defective product to the customer. Costs also occur if the model classifies defective steel product as normal ($fn$). Assuming there is no cost of true classification, the total cost of model misclassification is

$$Total\ cost = fp_{cost} \times fp + fn_{cost} \times fn$$ (6)

A. EVALUATION
The first evaluation was done with Data_1. Data_1 was collected between May 2017 and November 2019 and was divided into a training and test set at an 80%/20% division, respectively. The test part of Data_1 was collected between 27 June 2019 and 30 November 2019.

Figure 5 shows the ROC curve for dataset D11_test, with an AUC of 0.81. As an example, suppose the classification threshold is selected so that about half of the defects are found. In this case, we used a threshold that produces a 0.1 false positive rate ($fpr$). With this threshold, the MCC was 0.27. Multiple slabs are also produced during each individual heat, and some of the process measurements are made multiple times during slab production. These measurements can be aggregated and then used to derive new features, such as measurement mean value. The above-mentioned AUC and MCC values were thus measured using the minimum, maximum, and mean values of the aggregated measurements.

When the same classification model was tested with dataset Data_2 (collected between 1 December 2019 and 31 May 2020), the AUC was 0.71 and the MCC 0.29, with a threshold of 0.17. Figure 6 presents the ROC curve using Data_2. The AUC is lower than in Data_1, but the MCC is higher. This is due to the shape of the ROC curve, where the selected $fpr = 0.1$ produced a higher $tpr$ than in Data_1.

The performance analysis shows that the quality detection model could assist in quality inspection by proposing casted slabs into a further, more detailed manual inspection. Due to the large variations in defects and an otherwise challenging environment, it is often difficult to detect defects in the slab...
surface either manually or automatically with, for instance, camera-based systems. Manual inspection is labor-intensive, time-consuming, and prone to random errors as well [42]. The approach presented in this study allows manual inspection operators to spend more of their limited schedule inspecting slabs that are more likely to be defective. To concretize the quality classification performance provided by the model, an example threshold point can be selected from ROC curve. For example, by inspecting only about 10% of the slabs, more than 50% of the defects could be found when tested with the test part of $D_{11}$. Classification threshold level in the production environment can be later fine-tuned e.g. based on the cost of classification outcomes as presented in the equation 6.

Between Data_1 and Data_2, there was a degradation in prediction performance. For instance, the best AUC in the validation dataset from the validation procedure presented in Figure 3 is 0.83, with the AUC of Data_1 test set $D_{11}$ 0.81 and that of $D_{21}$ 0.71, so prediction performance deteriorated over time. Such behavior in a real-world environment is to be expected because the production environment is always non-stationary for various reasons. This kind of dataset shift means that training and test distributions are different—that is, $P_{\text{train}}(y|x) \neq P_{\text{test}}(y|x)$ and $P_{\text{train}}(x) \neq P_{\text{test}}(x)$. One potential reason for a concept drift may be manual quality inspection, which can vary greatly depending on, for example, inspector expertise and production environment. In addition to the covariate shift, this can be an interesting topic for future work.

B. MODEL EXPLAINABILITY

Shapley values can be utilized to analyze which features the model uses to decide between a normal and defective steel product. Table 6 shows the most important features for the $D_{11}$ training and test subsets to predict steel quality. Based on the Shapley values, the same features seem to explain the decisions made by the prediction model in both of these $D_{11}$ subsets. The mold surface maximum difference, heat transfer capability, and casting order were the three most influential parameters in both datasets as well.

Figure 7 visualizes the most important features based on Shapley value and their effect on model output. Red indicates a high feature value, and a higher output value represents the increased probability of a defective steel product. For example, based on Shapley values, high values in the mold surface maximum difference seem to correlate with a higher chance of surface defects. In addition, a lower mold heat
TABLE 6. Important features for the training and test parts of D_{11}.

| Rank | Important features: \( D_{11\_train} \) | Important features: \( D_{11\_test} \) |
|------|----------------------------------------|----------------------------------------|
| 1    | Mold surface maximum difference, mother slab mean value | Casting order |
| 2    | Mold heat transfer capability, mother slab minimum value | Mold heat transfer capability, mother slab minimum value |
| 3    | Casting order | Mold surface maximum difference, mother slab minimum value |
| 4    | Mold surface maximum difference, mother slab maximum value | Casting speed, mother slab minimum value |
| 5    | Casting speed, mother slab minimum value | Mold surface maximum difference, mother slab maximum value |
| 6    | Casting speed, mother slab mean value | Casting speed, mother slab mean value |
| 7    | Temperature target after ladle treatment | Temperature target after ladle treatment |
| 8    | Temperature target in tundish, mother slab mean value | Temperature target in tundish, mother slab mean value |
| 9    | Sulfur in the mold | Sulfur in the mold |
| 10   | Argon pressure | Argon pressure |

transfer capability correlates with a higher risk of surface defects.

By providing the insight and transparency on which the model has made the decision on the quality of the steel slab, the process engineer can gain confidence in the correctness of the decision. In addition to gained transparency, process engineers can utilize explanations to improve the process. Some of the process parameters (such as process temperatures) may be directly editable. If it is found that this kind of process parameter significantly affects the quality of the process, this process parameter can be directly modified. In the case of non-editable process parameters (such as slab casting order), process engineers and operators can invest more time in manual inspection of the slab to find possible defects.

VI. CONCLUSION

This paper presents a steel quality prediction system that can predict surface defects in steel products during manufacturing. A key component of this system is a gradient boosting-based model trained to classify steel products into two classes, normal and defective. This study used data from a real SSAB steel production facility located in Raahe, Finland with a 2.8-Mton crude steel production capacity. The model used data from the liquid steel phase with 89 input features and was able to make a quality prediction for each slab before sending them to the plate rolling phase. Reliably predicting production quality in the early phases of steel manufacturing can reduce manufacturing costs and save time if unnecessary production steps to address defective products can be avoided or if corrective actions can be taken for the product at an early production stage.

Another key aspect of this study is the improved transparency of the quality prediction model. Shapley values were used to calculate the importance of each input feature in the model predictions. Accordingly, our study revealed an explanation for each produced steel slab and thus how operators can utilize the model to understand which parts of the production process might have the greatest impact on current steel quality, if problems occur. In addition, the explanations can be used in the model validation process. The quality prediction model was integrated into QMT software, so process engineers in the manufacturing plant can easily review the process and how the model predicts quality.

The analyzed data were collected in two separate batches. The first dataset was historical data collected at the steelmaking plant from May 2017 to November 2019 and was used for both model training and initial validation. The second dataset was collected during the testing period, when the model was tested in the production environment as part of the QMT tool. The testing period started in December 2019 and ended in May 2020. The validation process showed that the prediction model can assist process operators in finding slabs with the highest risk of defects by locating 50% of defective slabs but only marking 10% of the slabs as potentially defective. This can greatly improve the efficiency of the steelmaking process.

REFERENCES

[1] C. C. Xu and D. Q. Cang, “A brief overview of low CO₂ emission technologies for iron and steel making,” J. Iron Steel Res., Int., vol. 17, no. 3, pp. 1–7, Mar. 2010, doi: 10.1016/S1006-706X(10)60064-7.
[2] M. T. Johansson and M. Söderström, “Options for the Swedish steel industry—Energy efficiency measures and fuel conversion,” Energy, vol. 36, no. 1, pp. 191–198, Jan. 2011, doi: 10.1016/J.ENERGY.2010.10.053.
[3] D. Kushnir, T. Hansen, V. Vogl, and M. Åhman, “Adopting hydrogen direct reduction for the Swedish steel industry: A technological innovation system (TIS) study,” J. Cleaner Prod., vol. 242, Jan. 2020, Art. no. 118185, doi: 10.1016/J.JCLEPRO.2019.118185.
[4] D. Weichert, P. Link, A. Stoll, S. Rüping, S. Hlenfeldt, and S. Wrobel, “A review of machine learning for the optimization of production processes,” Int. J. Adv. Manufact. Technol., vol. 104, nos. 5–8, pp. 1889–1902, Jun. 2019, doi: 10.1007/S00170-019-03988-5.
[5] J. P. Yun, S. Choi, B. Seo, C. H. Park, and S. W. Kim, “Defects detection of billet surface using optimized Gabor filters,” Tech. Rep., 2008, doi: 10.3182/20080706-5-KR-1001.4088.
[6] A. Landström and M. J. Thorley, “Morphology-based crack detection for steel slabs,” IEEE J. Sel. Topics Sign. Process., vol. 6, no. 7, pp. 866–875, Nov. 2012.
[7] V. B. Trofimov, “Automated expert systems in blast-furnace process control,” Metallurgist, vol. 64, nos. 1–2, pp. 3–12, Mar. 2020, doi: 10.1007/S11015-020-00961-6.
[8] T. Ko, J. H. Lee, H. Cho, S. Cho, W. Lee, and M. Lee, “Machine learning-based anomaly detection via integration of manufacturing, inspection and after-sales service data,” Ind. Manage. Data Syst., vol. 117, no. 5, pp. 927–945, Jun. 2017, doi: 10.1108/IMDS-06-2016-0195.
[9] A. Caggiano, J. Zhang, V. Alfieri, F. Caiazzo, R. Gao, and R. Teti, “Machine learning-based image processing for on-line defect recognition in additive manufacturing,” CIRP Ann., vol. 68, no. 1, pp. 451–454, 2019, doi: 10.1016/j.cirp.2019.03.021.
[10] A. Adadi and M. Berrada, “Peeking inside the black-box: A survey on explainable artificial intelligence (XAI),” IEEE Access, vol. 6, pp. 52138–52160, 2018, doi: 10.1109/ACCESS.2018.2870052.
S. Tamminen, H. Tiensuu, E. Ferreira, H. Helaakoski, V. Kyllönen, D. Laha, Y. Ren, and P. N. Suganthan, “Modeling of steelmaking process with effective machine learning techniques,” in Proc. Ind. Conf. Data Mining, Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 10777, 2018, pp. 17–28, doi: 10.1007/978-3-319-95786-9_2.

J. Zhang, D. Mucs, U. Norinder, and F. Svensson, “LightGBM: An effective and scalable algorithm for prediction of chemical toxicity–application to the ToxCast and mutagenicity data sets,” J. Chem. Inf. Model., vol. 59, no. 10, pp. 4150–4158, Oct. 2019, doi: 10.1021/acs.jcim.9b00633.

S. Guo, J. Yu, X. Liu, C. Wang, and Q. Jiang, “A predicting model for properties of steel using the industrial big data based on machine learning,” Comput. Mater. Sci., vol. 160, pp. 95–104, Apr. 2019, doi: 10.1016/J.COMMATSCI.2018.12.056.

D. Laha, Y. Ren, and P. N. Suganthan, “Modeling of steelmaking process with effective machine learning techniques,” Expert Syst. Appl., vol. 42, no. 10, pp. 4687–4696, Jun. 2015, doi: 10.1016/J.ESWA.2015.03.030.

S. Tamminen, H. Tiensuu, E. Ferreira, H. Helaakoski, V. Kyllönen, J. Jokisaari, and E. Puukko, “From measurements to knowledge—Online quality monitoring and smart manufacturing,” in Proc. Ind. Conf. Data Mining, Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 10933, 2018, pp. 17–28, doi: 10.1007/978-3-319-95786-9_2.

SSAB’s Annual Report 2018 Published—SSAB. Accessed: Nov. 1, 2019. [Online]. Available: https://www.ssab.com/company/about-ssab/ssab-in-brief/the-abc-of-steel

P. Storm, “Using expert systems in blast furnace operation—A few preliminary impressions,” in Strategic Management of the Manufacturing Value Chain. 1998, pp. 623–630, doi: 10.1007/978-3-387-35521-0_69.

A. Sorsa, J. Ruuska, J. Lilja, and K. Leiviskä, “Data-driven multivariate analysis of basic oxygen furnace used in steel industry,” IFAC-PapersOnLine, vol. 48, no. 17, pp. 177–182, Jan. 2015, doi: 10.1016/J.IFACOL.2015.10.099.
VESA KYLLÖNEN received the Graduate degree in information technology from the University of Oulu, in 1998. He is currently a Research Scientist with the VTT Technical Research Centre of Finland. Since his graduation, he has been working on several topics, such as data mining, scheduling, global optimization, route optimization, automatic calibration, shape optimization, bin packing, context awareness, machine learning, data visualization, and multi-user adaptation. His work also covers software design and implementation.

LEENA MÄÄTTÄ received the Graduate degree in process metallurgy from the University of Oulu, in 2000. She is currently a Senior Development Engineer at SSAB Europe Raahe Works. She has been working for several steelmaking unit processes in process development at Raahe Steel Plant, since 2001. In 2020, she began working as a Quality Manager at steel plant. Her research interests include quality improvement, good and productive management, and cooperation.

AGNE BOGDANOFF received the Master of Science degree in process automation from the University of Oulu, in 1993, and the Licentiate degree in technology, in 1998. She is currently a Process Development Manager at SSAB Europe Oy Raahe Works. She started at SSAB Raahe Steel Works, in 1998, in hot rolling mill development. She has worked as a Project Manager in several national funded projects and also on the RFCS projects “Reduction of Shape Defects and Yield Losses by Advanced Online-Adaptation of Control Systems and New Operating Strategies in Heavy Plate Rolling Mills” (SHAPEHPM) RFS-CR-03039 and “Optimization of Final Plate Flatness by Set-Up Co-ordination for Subsequent Manufacturing Processes” (FINALPLATEFLATNESS) RFSR-CT-2005-00019. From 2008 to 2015, she worked as a Production Manager at slab halls. Since December 2015, she has been as a Process Development Manager in steelmaking process development. At the moment, she is a Leader of process development at Raahe Steel Works. Her research interests include quality improvement, coaching, and cooperation development.

* * *