Improved Prediction of Steel Hardness Through Neural Network Regression

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.

Authors

Colin Elkin
Assistant Professor, Purdue University Northwest, Hammond, Ind., USA
cpe@pnw.edu

Rajat Bathla
Lead Engineer – HSM Quality, Cleveland-Cliffs Burns Harbor, Burns Harbor, Ind., USA
rajat.bathla@clevelandcliffs.com

Thomas Poplawski (pictured)
HSM Manager, Optimization, Cleveland-Cliffs Burns Harbor, Burns Harbor, Ind., USA
thomas.poplawski@clevelandcliffs.com

Soumitra Agashe
Senior Division Manager, Hot Rolling and Finishing Operations, Cleveland-Cliffs Burns Harbor, Burns Harbor, Ind., USA
soumitra.agashe@clevelandcliffs.com

Vijay Devbhaktuni
Professor and ECE Department Chair, Purdue University Northwest, Hammond, Ind., USA
vjdev@pnw.edu

Product hardness number is a vital input for finishing mill setup models in the calculation of load, torque and power in a hot strip mill. It is based on steel grade, whose value is updated from mill feedback. The goal of this work is to improve hardness prediction even if the steel grade is new to the hot mill. To do so, a robust neural network–based hardness prediction algorithm is proposed that inputs statistically significant features such as chemistry composition, aim gauge, aim width, rolling temperature and finishing temperature. Ultimately, this approach achieves mean absolute error below 2.1%, which substantially outperforms the baseline error of 2.75%.

Predictive Models for Finishing Mill

Within the finishing mills and hot rolling processes, hardness is a steel property of great significance that is influenced by factors such as temperature[1] and its chemical composition.[2] Having the ability to predict qualities such as hardness can be highly beneficial in the design of new products and for analyzing rolling schedules for schedule consolidation. Researchers have previously sought similar efforts in hot mill operation, such as prediction of coiling temperature[3] and detection of surface defects.[4]

In recent years, a variety of new mill technology has been made available to aid in similar processes such as inspecting the surfaces of rolls[5] and producing thinner roll strips.[6] The vast focus of such technological advancements, however, lies within the emergence of Industry 4.0. Industry 4.0 brings about a plethora of new technologies and industrial concepts for the steel industry to explore and bring to new and existing processes. For instance, a technology known as condition-based monitoring[7] uses mill data to monitor the various conditions of processes and alerts operators when something is running outside of a normal set of constraints, such as anomalies in the pipeline or maintenance and safety needs.

An increasingly vital category of industrial technology lies within artificial intelligence (AI), or more specifically the subset of AI known as machine learning (ML). A variety of ML techniques have been applied to different steel applications so far, such as image processing for detection of surface defects using convolutional neural networks[8] and the use of knowledge discovery from data (KDD) to form an entire data processing pipeline for predictive or explanatory models.[9] Other methods include the use of self-organizing maps (SOMs) for the discovery of disuniformities[10] as well as support vector machines (SVMs) for temperature prediction.[11]

Overview of Neural Network

This paper presents a feedforward artificial neural network (ANN) to predict product hardness given a variety of inputs, as illustrated in Fig. 1. The inputs are a combination of chemistry composition, such as carbon, manganese and phosphorus, and non-chemistry features, such as aim gauge, aim width and temperature hardness. The product hardness is the sole output.

To determine an accurate ML model for hardness prediction, a supervised learning approach is employed using a feedforward ANN.

Figure 1

Black box diagram of inputs and output for artificial neural network (ANN).

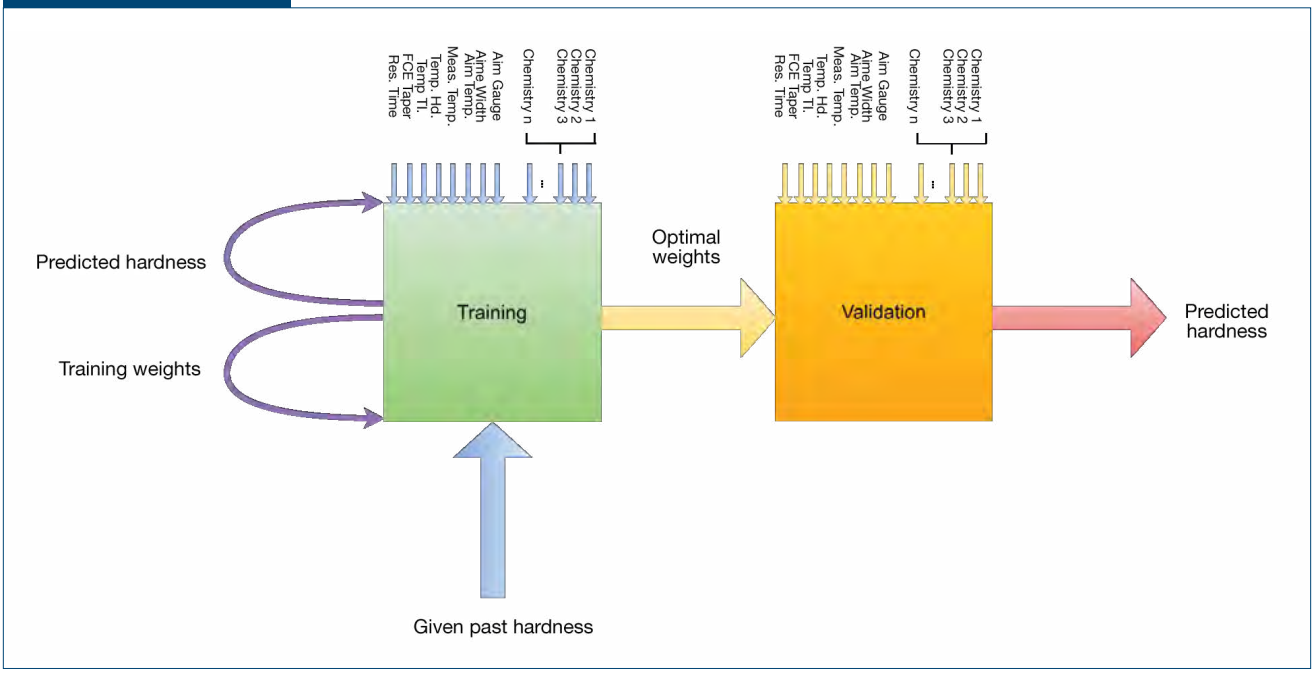
In supervised learning, data samples are shuffled and separated into training and validation data, each of which comprises the two different stages of the ML process. In the training stage, the inputs and outputs of the training data are given, and through iterative training the weight variables in the ANN equations are continuously updated until the percent error between the actual outputs and the predicted outputs are below a certain threshold.

In the validation stage, the weights are set to those determined at the end of the training stage, and only the inputs of the validation data are given. This is to verify that the neural network model can make accurate predictions without the outputs being known. This will be particularly useful if the algorithm is implemented into a real-time system in which the product hardness must be predicted before the actual value is known.

Fig. 2 provides the overall supervised learning process in terms of training and validation.

Upon the formation of this approach, the next challenge lies within deciding how to partition the

Figure 2



Overview of the supervised learning process.

training and validation stages. Because the goal of this research is to predict product hardness even when the steel grade is new to the hot mill, it was decided that the most ideal approach is to use data samples with known steel grades as the training data and those with unknown grades as the validation data.

Next, the rest of the experimental parameters must be determined, the two most crucial of which are the ANN training method and the activation function. The training function is the algorithm that sets the weights to values that determine the best accuracy, while the activation function determines whether to activate certain neurons in the ANN based on the relevance of a neuron's input in determining an accurate output.

This work employs three different training methods and two activation functions. The training algorithms selected for experimentation are limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS), stochastic gradient descent (SGD), and a variation of SGD known as adaptive moment estimation (Adam), while the activation functions are sigmoid and rectified linear unit (ReLU).

Grade-Based Clustering

To further enhance hardness prediction accuracy, a clustering portion was added to the overall prediction process. Clustering is a form of unsupervised machine learning in which outputs are predicted based on patterns in the input data, rather than by training of some prior data. Here, clustering is used to separate data samples based on the characteristics of different steel grades. By isolating different sets of data into ones with either a single steel grade or multiple similar steel grades, ideally, more accurate weights can be obtained by having a more focused data set, thereby forming more meaningful and more accurate hardness predictions.

In this approach, K-means clustering is used, which partitions data samples into K clusters with the closest mean. In this approach, the full data set (including known and unknown steel grades) is taken, dimensionality reduction applied via principal component analysis (PCA), and then the K-means clustering algorithm is applied with a pre-determined number of clusters. From there, each cluster is a separate data set to be used in its own supervised learning process, again with the known steel grades as training and the unknown grades as validation. This way, each cluster has an ANN whose weights are dependent on the given steel grades.

In this research, the ANN is evaluated in terms of both clustered and unclustered data.

Initial Neural Network Results

To thoroughly verify the effectiveness of the hardness prediction algorithms, the neural network was tested under a variety of experimental parameters. These include:

- ANN training technique.
- Activation function.
- Training versus validation.
- Number of clusters.

To evaluate the effectiveness of the methods used in this study, mean absolute error is used, which is defined as:

$$\text{Error} = 100 \times |y_{\text{actual}} - y_{\text{predicted}}| \; / \; y_{\text{actual}}$$

averaged across all data samples for either the training set or the validation set, depending on the type of accuracy being determined. Based on efforts in the authors' prior work, the baseline error that was intended to undercut is 2.75%. Table 1 presents the ANN results for each combination of training method and activation function in terms of training and validation accuracy.

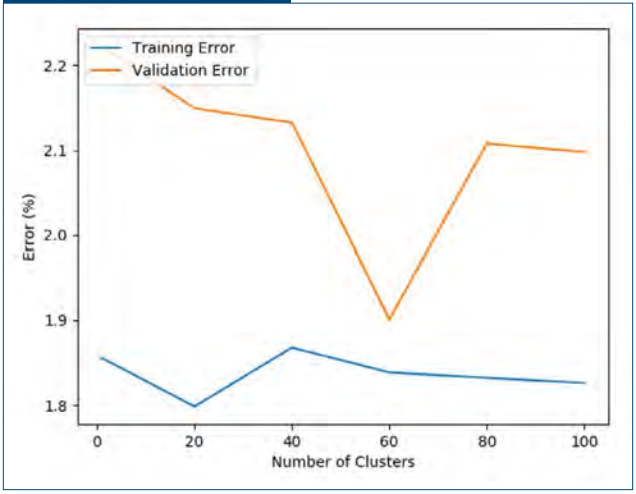
Table 1

Overview of Training and Validation Error for Each Combination of Training Method and Activation Function

	Training error (%)		Validation error (%)	
	Sigmoid	ReLU	Sigmoid	ReLU
L-BFGS	1.92	1.92	3.43	6.42
SGD	2.58	2.58	2.91	16.30
Adam	2.32	1.68	4.99	3.65

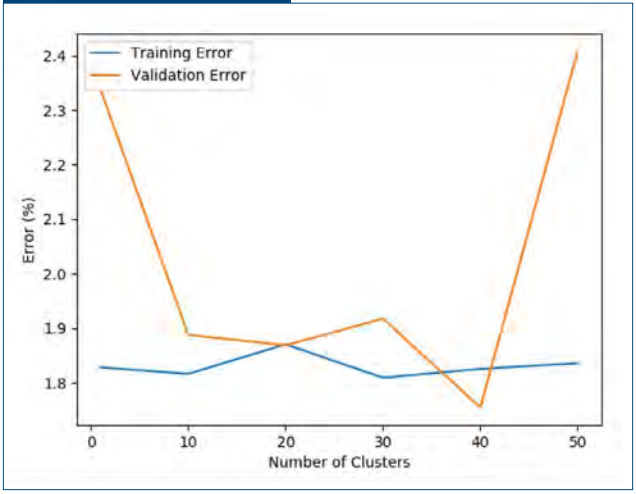
As the results in the table indicate, for the training accuracy, L-BGFS produces the lowest error while SGD performs the worst, and the difference in results between activation functions is negligible. However, under validation error, a combination of sigmoid and SGD provides the best results. While this corresponds to the highest training error, that is less relevant, as a relatively minor difference between error values is preferred in supervised learning. In the case of L-BFGS and ReLU, the smallest training error comes at the expense of one of the highest validation error values. This is most likely due to overfitting, in which the data (known steel grades) is trained too well, with an insufficient amount of validation data (unknown grades) to replicate similar results in the validation stage. Due to the overall success of SGD and sigmoid, this training method and activation function will be

Figure 3



Plot of accuracy vs. number of clusters in increments of 20 clusters.

Figure 4



Plot of accuracy versus number of clusters in increments of 10 clusters.

exclusively used for the remainder of the experimental results.

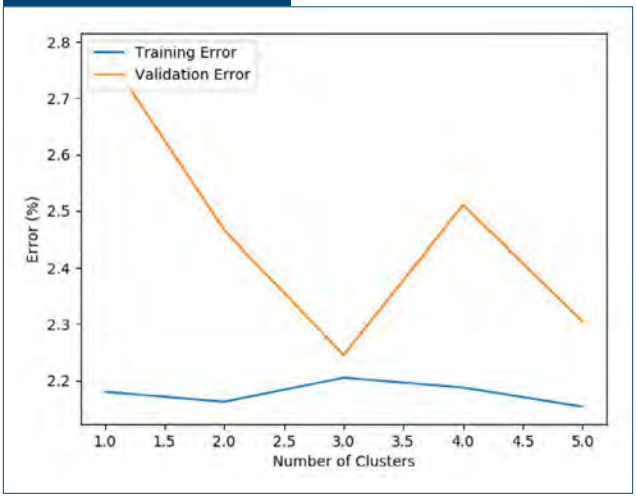
Figs. 3–5 provide the results under varying numbers of clusters, as described in the previous section. To provide a more thorough picture of how clustering affects the accuracy results, three different sequences of cluster amounts are used, ranging from coarse (varying by 20) in Fig. 3 to medium (varying by 10) in Fig. 4 to fine (varying by 1) in Fig. 5.

Fig. 3 plots the training and validation error versus the number of clusters. As it can be seen, the training error has little variation between cluster amounts, ranging from 1.8 to 1.9% the entire time. The validation error, however, has a clear minimum at 60 clusters with 1.9%, while most other clusters have error ranging from 2.1 to 2.2%. Fig. 4 provides a similar plot but with clusters varying by 10 instead of 20 (i.e., medium tuning rather than coarse tuning).

In this case, the training error retains similar values, while in the validation error, there is a global minimum of 40 clusters with less than 1.8% error as well as a local minimum at 20 clusters. It is worth noting that the values at 20 and 40 differ somewhat from those in the previous figure. However, by applying a margin of error of 0.4%, better consistency can be achieved. However, once doing so, all values in Fig. 3 are now within the margin of error, thereby leaving the use of clusters as statistically insignificant in terms of improving accuracy. Luckily, this is not the case in Fig. 4, as 1 and 50 clusters are essentially the outliers. Lastly, one additional plot provides results under fine tuning of the number of clusters.

In this case, the training error still holds steady, hovering around 2.2%. The number of clusters appears to be optimized at 3.0, with all other values rising as high as 2.8%. In addition, all cluster amounts except 1 are

Figure 5



Plot of accuracy versus number of clusters in increments of one cluster.

within the margin of error. When analyzing all three plots overall, it can be found that the separating different steel grades into clusters of an amount between 1 and 40 does in fact improve prediction accuracy. However, more experimental parameters can still be explored to achieve the optimal hardness prediction accuracy.

Feature Selection

Upon initial interpretation of the results, a key observation to consider is that with a total of 21 inputs, or features being fed into the ANN, it must be considered

whether every feature is actually useful in obtaining an accurate hardness prediction. Furthermore, in the case of features that are beneficial, which ones are more crucial than others? This is the case in which the concept of feature selection becomes of vital importance.

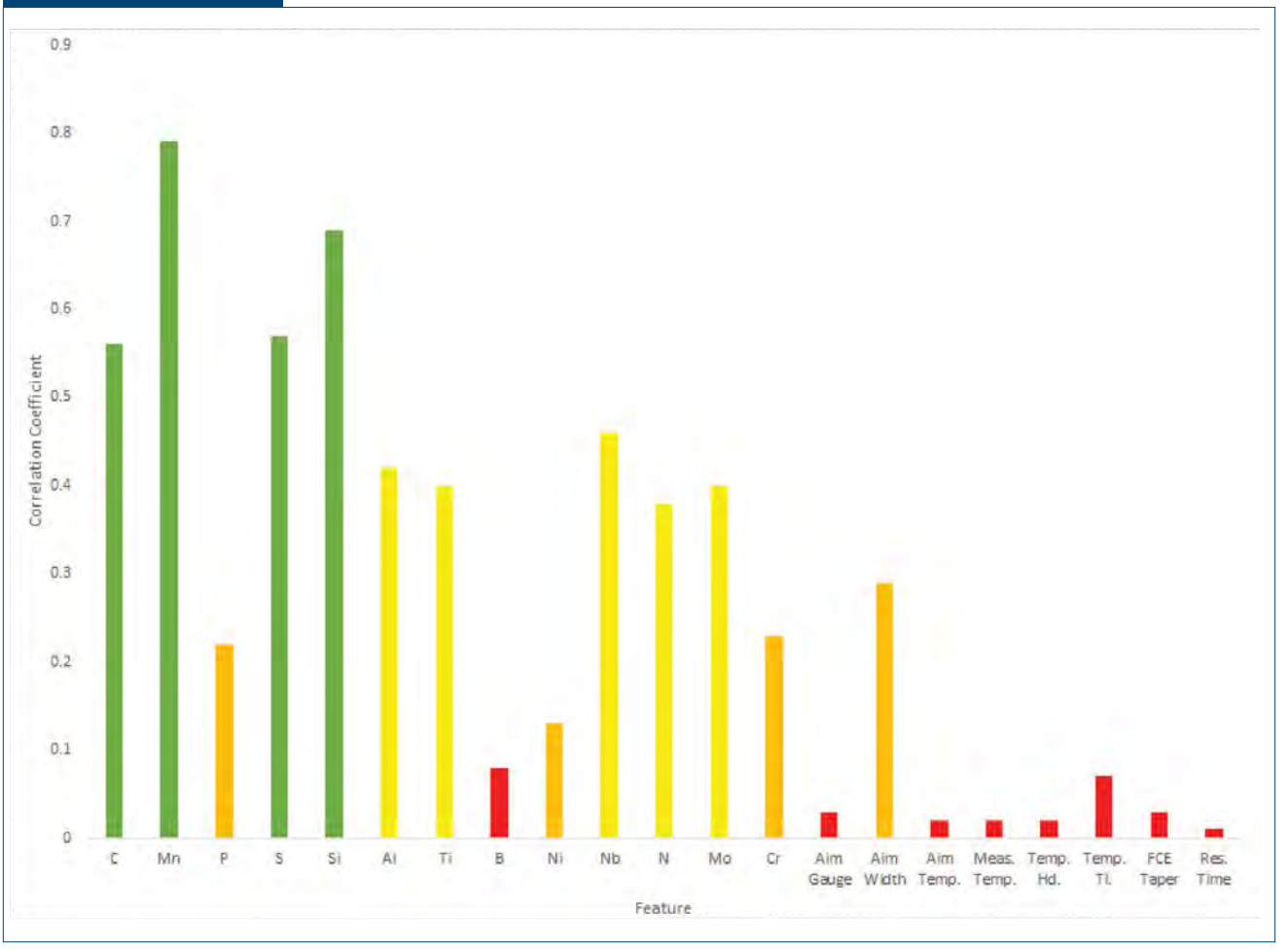
Feature selection is the process of selecting a particular set of inputs that has the most relevance in determining the output. Essentially, the features with the highest amount of correlation with the output are of interest. To determine such an optimal set of features, there are three possible ways to achieve this. One is a filter method, in which statistical processes can be applied to filter the features that pass a certain threshold before moving on to the ML process. Another is a wrapper method, in which the ML process is executed repeatedly with multiple different combinations of features. A third approach is an embedded method, in which the optimal subset of features is determined while the ML process is executed.

In this case, the filter method was selected, in which the correlation coefficient between each feature and the output hardness are used to determine the relevance of said feature. The correlation coefficient is defined as the covariance between a feature and the output divided by the product of the standard deviation of each. Fig. 6 provides the correlation results for each feature.

In this figure, each feature can be classified into four groups: great correlation, denoted by green in which the coefficient exceeds 0.5; good correlation, of 0.3–0.5 range in yellow; fair correlation, of 0.1–0.3 range in orange, and poor correlation, denoted by red in which the coefficient is below 0.1. In many other applications, the standards for good correlation are typically much higher, such as 0.8 or 0.9, but in this case, the features evaluated relative to one another rather than on a fixed scale.

This relative analysis only shows features that meet the top threshold and all happen to be chemistry-based, which are carbon, manganese, sulfur and

Figure 6



Comparison of correlation coefficients for each available feature.

Table 2

Validation Error (%) by Feature Set (organized by correlation threshold) and by Number of Clusters

	No. of clusters					
Feature Set	1	10	20	30	40	50
CORR(Xi,Y) > 0.5	2.30	2.34	2.32	2.48	2.17	2.39
CORR(Xi,Y) > 0.3	1.72	1.78	1.60	1.63	1.64	1.78
CORR(Xi,Y) > 0.1	2.36	2.42	2.22	2.66	2.15	2.04
CORR(Xi,Y) > 0	2.34	1.89	1.87	1.92	1.76	2.41

silicon. The next highest tier is also entirely chemistry-based, containing aluminum, titanium, niobium, nitrogen and molybdenum. The fair tier has only one non-chemistry-based feature, which is aim width. The other features are phosphorus, nickel and chromium. Finally, the lowest tier contains only one chemistry-based feature (boron) and all non-chemistry-based ones except for aim width. Thus, overall, it can be concluded that chemistry composition in general has the strongest correlation with product hardness.

Because none of the features have a correlation that exceeds 0.8, rather than run the neural network with only the strongest correlated features, it was felt that it would be more beneficial and explanatory to run the algorithm with each of the four thresholds, ranging from all features to only the features with the strongest correlation, applying incremental correlation thresholds of 0, 0.1, 0.3 and 0.5, respectively. Like before, the algorithms were conducted as the averages of 10 independent trials. Based on the results in the previous section, it was decided to use SGD as the training technique and sigmoid as the activation function.

Table 2 provides the accuracy of the results, where the rows correspond to the correlation thresholds, while the columns correspond to different numbers of clusters. Here, the error is lowest under the second most restrictive set of features and with 20 clusters. In addition, under every cluster in this row, the results are below 2.1% and thereby substantially outperform the baseline error of 2.75%.

Conclusions

This paper presented a neural network–based hardness prediction algorithm that inputs features such as chemistry composition, aim gauge, aim width, rolling temperature and finishing temperature. The overarching goal was to improve hardness prediction for both known and unknown steel grades. While applying a supervised ANN approach using known label grades as training data and unknown grades as validation, a mean absolute error of well below 2.1% was achieved, following extensive experimentation with combinations of activation functions, training algorithms, cluster sizes and correlation thresholds. Ultimately, this substantially outperformed the baseline error of 2.75%.

For future work, the most imminent next step is to implement this algorithm in a real-time system using embedded C code. In addition, it is desirable to utilize these algorithms to solve complex steel industry problems such as improving shape from the hot strip mill and increasing mill productivity by reducing number of rolling schedules.

References

1. N. Yukawa, E. Abe and S. Fujiwara, "Thermal Properties of Oxide Scale on Surface of Work Roll in Hot Rolling Mill," *Procedia Manufacturing,* Vol. 15, 2018, pp. 59–64.
2. J.C. Herman, "Impact of New Rolling and Cooling Technologies on Thermomechanically Processed Steels," *Ironmaking & Steelmaking,* Vol. 28, No. 2, 2001, pp. 159–163.
3. H.B. Xie et al., "Prediction of Coiling Temperature on Runout Table of Hot Strip Mill Using Data Mining," *Journal of Materials Processing Technology,* Vol. 177, No. 1–3, 2006, pp. 121–125.
4. H. Jia et al., "An Intelligent Real-Time Vision System for Surface Defect Detection," *Proceedings of the 17th International Conference on Pattern Recognition,* 2004, pp. 239–242.
5. B. Lopez and J. Baczynsky, "The Next Generation of Mill Roll Inspection Systems: A Case Study of a New Inspection System," *AISTech 2019 Conference Proceedings,* 2019, pp. 2061–2067.
6. A. Jungbauer et al., "Arvedi ESP: Reaching Maturity in Endless Casting and Rolling Technology," *AISTech 2019 Conference Proceedings,* 2019, pp. 1537–1543.
7. C. Cotton, "Industrial Change: Impact of Emerging Technologies on the Steel Industry," *AISTech 2019 Conference Proceedings,* 2019, pp. 2619–2627.
8. G. Pellegrini et al., "Successful Use Case Applications of Artificial Intelligence in the Steel Industry," *AISTech 2019 Conference Proceedings,* 2019, pp. 2573–2584.
9. M. Sapparat et al., "Driving Business Value in Industry 4.0: Big Data and Analytics for the Improvement of Steel Processes," *AISTech 2019 Conference Proceedings,* 2019, pp. 2709–2719.
10. G. Nastasi et al., "SOM-Based Analysis to Relate Non-Uniformities in Magnetic Measurements to Hot Strip Mill Process Conditions," *Multidisciplinary Approaches to Neural Computing,* Springer, 2018, pp. 223–231.
11. M. Luccini et al., "Intelligent Steelmaking Based on Advanced Analytics: Reducing Operational Costs of a BOF," *AISTech 2019 Conference Proceedings,* 2019, pp. 2593–2601. ✦

This paper was published in the AISTech 2020 Conference Proceedings. AIST members can access the AISTech 2020 Conference Proceedings in the AIST Digital Library at digital.library.aist.org.