Smart Ladle: AI-Based Tool for Optimizing Caster Temperature

Nicholas J. Walla, Zhankun Luo, Bin Chen, Yury Krotov, Chenn Q. Zhou

Center for Innovation through Visualization and Simulation
Purdue University Northwest
2200 169th St, Hammond, IN 46323, USA
Phone: 1-219-989-2765
Email: civs@pnw.edu

Steel Dynamics, Inc.
4500 Country Road 59, Butler, IN 46721, USA

Keywords: Machine learning, Refining ladle, Ladle heat loss, Continuous casting, Smart manufacturing

ABSTRACT

From the BOF/EAF to the caster, the ability to quantify and respond to the variables that affect steel casting temperature is crucial for achieving consistent casting quality and maximizing productivity. Deviations from the optimum steel casting temperature can require adjustment to casting speed, which impacts productivity and can also harm product quality. This work will use a deep-learning network to develop quantifiable relationships between the casting temperature and various factors during the ladle refining process to enable predictions of casting temperature and precise adjustments to steel temperature prior to the ladle reaching the casting stage of the production process.

INTRODUCTION

The development of the Smart Ladle focuses on taking data collected from the ladle process and creating history-based predictions for ladle heat loss and tundish temperature behavior, then providing these predictions to operators so that they can make process decisions with better information. There are varying approaches to solving this issue, including the approach of this work. Operators can only react on information they know; the LMF operator knows to expect additional heat loss in the first few heats of a ladle's campaign, but they may not know about the long wait at the tap car that allowed a well-used ladle to cool. Data collection and presentation gives the operator more knowledge to work with, but still there exist hidden correlations between the data.

A key part of modern-day manufacturing across all sectors is the increased implementation of data collection systems and software utilities that make use of the collected data. This is especially true for the steel industry, where the progress of *smart manufacturing* and *Steel 4.0* use modern technologies to reduce costs and improve product quality.[1] One such technology that is having impact on many industries is machine learning and more recently deep learning.[2][3]

Data usage may be as simple as "gather and display", giving operators and technicians access to data feeds that help inform operations and design. Collected data can be further used with control systems to improve automation and safety, using data feeds to augment process rates or initiate emergency procedures and alarms. Where these two concepts meet sits deep learning: the use of data history, real-time feeds, and "fuzzy-logic" algorithms to create correlations in the data feed that allow for process optimization and prediction.

Advancements such as these allow for greater consistency and higher production quality, ensuring that operators have the information they need to apply expertise while filling in knowledge gaps with algorithm-based decision making. In the continuous casting process, operator expertise plays a key role in balancing casting parameters using the knowledge of current process conditions. Cast too slowly with low temperatures and you risk clogging, cast too quickly or with high temperatures and a breakout may occur.

Ladle Process

Referring to the "ladle process" is in may not be specific enough given the wide variety of logistical and procedural differences between different steel manufacturers. For the initial stages of this work, the ladle process at SDI Butler Division was used for testing and development. The ladle history data mentioned above is gathered from the following stations (roughly in order):

- Ladle preheating
- Electric arc furnace (EAF)
- Ladle metallurgy furnace (LMF)
- Casting
- Various ladle maintenance stations

Some ladle processes have alternate stages than the above (such as basic oxygen furnaces in place of EAF or ladle treatment stations instead of LMF), and others have additional stations such as vacuum degassing. The final outcome of this work is to create a universal system that can handle these differences while still providing predictions of the ladle thermal behavior. To this end, the machine learning efforts must be organized in such a way as to understand the incoming data semantically (with respect to the data's *purpose*).

For this work, the different stages of the ladle's process are categorized by the "type" of time. These are separated into preheat time, wait time, residence time, and maintenance time. The wait time and maintenance time are considered "empty" time, where the ladle has no steel and therefore will rapidly lose heat to the environment. However, there is a difference between a long maintenance time prior to preheat followed by a short wait time at the furnace and the reverse scenario. Therefore, these two "empty" times are treated separately from each other. Figure 1 below shows a diagram of a ladle's process and the different divisions of time.

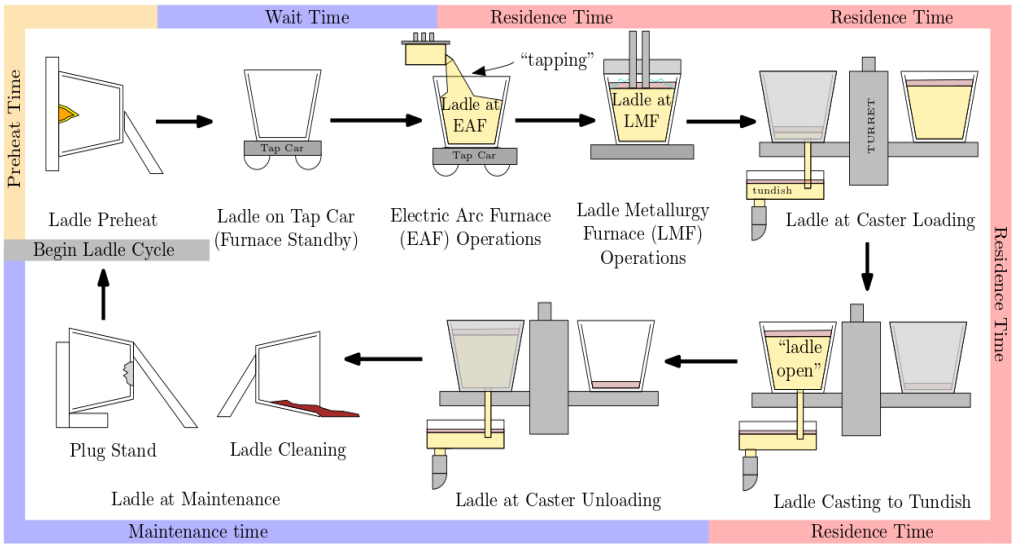

Figure 1. Ladle process diagram with different time categories.

Machine Learning

Machine learning is a field of computer science that revolves around the use of algorithms with overlapping (or "fuzzy") behavior rather than the discrete behavior found in most computer mathematics. Machine learning though methods such as neural networks (NN) creates those semantic correlations between the data input, finding connections that may not otherwise be apparent. For example, there is an intuitive connection between heat loss in the walls of a ladle and the time that a ladle spends empty (empty ladles lose heat to the environment, and there is no steel to provide replacement heat to those walls).

Engineers and technicians can take samples of ladle temperature to observe such a heat loss, and they may also take time samples to correlate this heat loss with time. This, however, requires the engineer to be aware of this correlation beforehand to manually make those connections. Machine learning takes broad swaths of data and examines each input relative to the desired outputs. The end result: correlating weights that describe the influence of each input on those outputs. These weights may be very small as to denote no correlation, or they may be comparatively large and indicate a strong direct connection (as in the time vs. heat loss example above).

The use of machine learning requires the proper "pruning" of the data to create a data structure: a properly-formatted input/output model for the machine learning algorithm. The algorithm itself can vary greatly. Some algorithms may be more applicable to short-term data feeds where most relevant data is immediately available. Others function in a way that allows the

algorithm to "remember" important data over a long history. Selection of an algorithm is as important as the creation of the data structure.

Once the algorithm has been prepared and the data put into the desired structure, the algorithm can be "trained" on the data. For machine learning, the more data that can be provided the neural network the better the network can learn. Longer data history ensures provides more chances for the data to include non-standard situations. If the neural network is only provided normal data, it will be unable to recognize and make prediction from abnormal inputs.

APPROACH

This work uses several assumptions and specific approaches to correlate the available input data to the desired outputs. Assessments and studies on the heat loss in ladle systems have been done to identify the major factors that influence the heat losses of the ladle and the correlations with tundish temperatures. Recommendations from industry partners and literature[4][5] resulted in a list of "critical parameters": factors of the refining ladle/casting process that will influence the thermal behavior of the ladle and are captured by the data available. These parameters of ladle history are:

1. Ladle history – data collected on a per-ladle basis, using the previous three heats of the ladle
    a. Ladle empty time – the time that the ladle has spent empty of steel, specifically between the end of casting and the start of preheating.
    b. Ladle preheat time – time spent on the preheating unit.
    c. Ladle steel time – time the ladle spends with liquid steel contact.
    d. Ladle gap time – time between the end of preheating and tapping of steel into the ladle.
    e. No. of heats in the campaign – the number of heat cycles the ladle has experienced since being relined. The first few heats of a new campaign will have different thermal behavior than a well-used ladle.
2. No. of heats in the sequence – the tundish itself is relined periodically, with the first heat of a new tundish having very different behavior
3. LMF temperature sample – the ladle steel temperature value manually sampled at the LMF
4. Casting throughput – the casting speed, necessary for predicting time-to-open and making assumptions on the future heat's performance
5. Tundish steel temperature – temperature samples of the steel temperature in the tundish

As mentioned earlier, some of the listed data is chosen because it is available. Previous studies in this field have found additional factors that will influence the heat loss in the ladle which are not currently available for usage, including ladle wall thickness/erosion, ladle wall temperature values, and the usage of ladle lids.[6][5] The model as developed can be modified to include such information if/when that information is available. Development of the software was done using Python 3.8 with the *numpy*, *scipy*, *pytorch*, and *pandas* libraries. For the training phase, a Linux machine with GeForce GTX 950M GPU was used. For simulating the SQL database calls that would occur in the production environment Microsoft SQL database, a similar MySQL database was built on the Linux machine using XAMPP. After the installation of the ODBC driver and MySQL ODBC connector, communication was established between the MySQL database and python program with the Pyodbc library.

The goal of the model is to make three predictions:

1. The ladle steel temperature drop between the last temperature measurement at the LMF and the time at which the ladle opens at the caster.
2. The tundish steel temperature at the midpoint of a heat
3. The slope of the tundish temperature profile after the linear region created by intermixing.

Providing these three predictions, in addition to providing the operator with quantitative and qualitative information about a ladle's history, will help operators with the decision-making process with regards to ladle steel temperature.

METHODOLOGY

Input Data

The SDI Butler Division in Butler, IN provided the input data for the model development. The collected data features mixed sample rates, with most of the data being event-triggered or manually triggered and others collected at fixed periods. The industry data was categorized in the following ways:

Table 1. Industry data used for model development.

Dataset	Description	Relevant Data
EAF Heats	Collated data on steel tapped into the ladle each heat	Ladle number, heat number, tap temperature, tap weight, time stamps
Ladle Event	Ladle repositioning data that is event-triggered	Ladle number, ladle location, time stamps, heat number
LMF Event	Data from the LMF process, event-triggered	Ladle number, arcing events, stirring events, ladle steel temperature samples, time stamps
Caster Process	Data on casting process and tundish, captured every five seconds	Ladle number, caster number, heat number, tundish steel temperature, ladle steel weight, tundish level, casting speed

The development of the model used offline copies of the database, with a roughly eight months of data (over 12000 heats) exported into a comma-separated format. After testing and training using the static data was completed, the code was modified to access an SQL database directly to read the necessary process data. This was tested by converting the exported data into a local SQL database that mirrored the database of SDI Butler Division and could be accessed for testing the database calls and local data analysis.

The pyodbc library was used to connect a Microsoft SQL database with the python program. After pulling data from the necessary tables, the input parameters can be parsed and the data structure prepared. For example, the remaining weight in an open ladle and the current throughput at that caster are used to calculate the expected time that the next ladle will be needed at that caster. This process checks for certain conditions that would prevent proper prediction. For example, when the specified ladle is not in the LMF process, or it does not yet have a temperate sample recorded, the python program would exit with comment. The software will also return an error if the heat currently casting has not been casting for long enough (generally at least half-complete).

Data Structure

The exported data must be parsed and formatted as inputs to the neural network. The resulting data structure includes 25 input factors: time intervals for ladle history, final LMF/tundish temperatures, casting throughput, and current heat data. Before feeding the data to neural networks, we must process industry data with normalization. For the purposes of training the algorithm, the data was filtered to exclude extreme "non-standard" heats. These included the first heat of a campaign, the first heat after downtimes/outages, and heats with values above/below certain extremes. This was to ensure that the resulting model would be trained using "standard" heats while additional methods for recognizing and processing non-standard heats were developed.

Software Integration

As the Smart Ladle software will need to work with different systems at different steel production facilities, development of the software included consideration for the differences in data security policies that may exist. Additionally, as the software was developed using python, further consideration was given as to the issue of preventing version conflicts and library management. As such, the *conda* environment was chosen for the deployment of the Smart Ladle software at SDI Butler Division. *Conda* creates a sandboxed virtual environment for the python distribution, as well as featuring a package management system that allows for easily installation of necessary python libraries in addition to version management. As such, the only software that needs to be installed on the target machine is the *microconda* environment, after which a batch script can be ran once to setup the necessary python version and libraries. This prevents the required python installation for the Smart Ladle from interfering with/overwriting other python installations.

To maintain this hands-off approach, the software is designed to interface with the *Ignition* system used by SDI Butler Division for their HMI system. A script was created for Ignition that will pass the necessary information to a batch script (ladle number and target caster), then execute the python code. The python code will then read the SQL database in a read-only configuration, process the data and run the algorithm to generate the predictions. The output from this command is then read by the Ignition script as an array, allowing it to be incorporated into the SQL database or an HMI panel as needed.

Machine Learning Algorithms

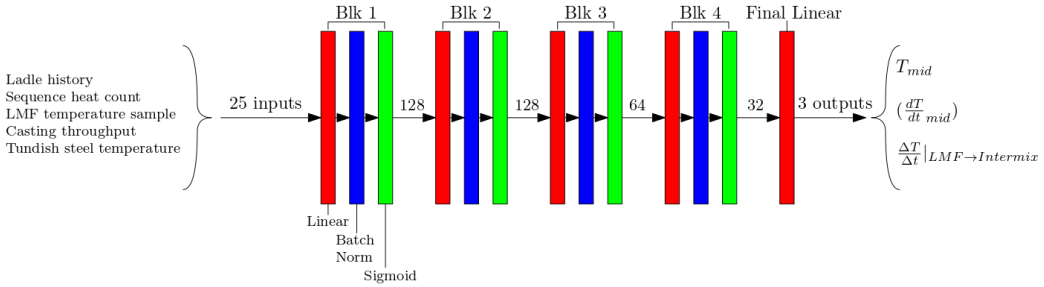

Figure 2. Architecture for Smart Ladle network.

The deep-learning program was developed to read inputs (process data) and provide outputs (ladle and tundish temperature predictions). The neural network architecture is demonstrated in Figure 2. It consists of 4 blocks for demonstrating input data and the final linear fully connected layer. At each block, we applied a linear layer to extract features, followed by a 1-Dimensional batch normalization layer and Sigmoid. The batch normalization layer[7] is necessary to reduce internal covariate shift and maintain the distribution of the inputs of each layer to produce reliable neural networks. Sigmoid is used to introduce nonlinearity to the neural networks. Next, we initiated the weights for each linear layer with Kaiming's method[8] due to the consideration of avoiding the vanishing gradient problem and exploding gradient problem. At the final layer, we used the fully connected layer to map 32 component features to the last three outputs.
Linear layer is expressed in following formula,

$$Y_m = \sum_{l=1}^{n_{in}} w_{l,m} Z_l + b_m$$

Batch normalization is defined as below,

$$\mu_B = \frac{1}{N}\sum_{i=1}^{N} x_i \, , \sigma_B^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_B)^2$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, y_i = \gamma \hat{x}_i + \beta \stackrel{\text{def}}{\equiv} \mathrm{BN}_{\gamma,\beta}(x_i)$$

Sigmoid is a simple nonlinear function,

$$Z_m = \left(\frac{1}{1 + e^{-Y_m}}\right)$$

Kaiming's Initialization can stabilize the stabilize the covariance of variable distribution,

$$\mathrm{var}[Y_m] = \mathrm{var}[\sum_{l=1}^{n_{in}} w_{l,m} Z_l] = \sum_{l=1}^{n_{in}} \mathrm{var}[w_{l,m}] \frac{1}{2} \mathrm{var}[X_l] = \frac{1}{n_{in}} \sum_{l=1}^{n_{in}} \mathrm{var}[X_l]$$

For the consideration of introducing nonlinearity in neural networks, the activation function plays a critical role. We applied the Adam algorithm[9] to reduce our loss function, which is a robust and widely used algorithm to find the minimum. It accelerates the process of convergence compared to traditional SGD method. It updates the weights of the whole neural networks after computing the gradient for a random sample:

$$w := w - \eta \frac{m}{\sqrt{v} + \epsilon}$$

For the loss function, we chose Smooth L1 loss (Huber Loss). Compared with the MSE loss function, it is less sensitive to significant errors that characterize MSE. Mainly, in some cases, it could prevent the exploding gradient problem to some extent.[10] Hence, it avoids excessive sensitivity to significant errors that characterize MSE.

$$l_i = \begin{cases} [f(x_i) - y_i]^2 & |f(x_i) - y_i| < 1 \\ |f(x_i) - y_i| - 0.5 & \text{otherwise} \end{cases}$$

$$loss(x, y) = \frac{1}{N}\sum_{i=1}^{N} l_i$$

Table 2 below shows a summary of the parameters for the developed machine learning model.

Table 2. Hyper-parameter values of the proposed model.

Hyper-Parameters	Value
Learning method	Adam
Loss function	Smooth L1 loss (Huber loss)
Activation function	Sigmoid
Linear layer initialization method	Kaiming's method
Initial weight distribution	Normal Distribution
Batch size	6
Learning rate	0.005
Iteration number	10,000
Linear layer number	5
Linear layer channel	{25×128; 128×128; 128×64; 64×32; 32×3}

Model Outputs

For the outputs of the neural networks, the target goals are the prediction of the temperature drop between the last LMF temperature sample and ladle open, to forecast the tundish temperature at the midpoint of the casting process, and to predict the slope of temperature changes in the linear region after intermixing has occurred. The outputs of the software include an array containing the three aforementioned values, as well as history information for the ladle's current heat and the three prior heats. Finally, an additional set of values categorizes these values with respect to a "typical" heat, allowing the operator to see both qualitatively and quantitatively the history of the ladle.

Table 3. Output array

Output Value	Description
Ladle data array	Heat #, ladle #, location, # heats in campaign, etc.
Midpoint temperature prediction	Numerical value [°F]
Midpoint slope prediction	Numerical value [°F/s]
Heat loss (LMF to Caster)	Numerical value [°F]
Ladle history array – Current heat (heat n)	Steel time, empty, time, wait time, preheat time
Ladle history arrays – Previous heats (heat n-1 to n-3)	As above
Ladle history values – Qualitative values	Qualitative labels for current heat and three previous
Comment	Comments on special scenarios or errors

The qualitative labeling is done by comparing the current value (e.g. "steel time") and comparing it to the standard deviation for the collection of all "normal" heats in the dataset. Then, labels are assigned based on this comparison:

Table 4. Qualitative ladle history value labeling.

Value Comparison	Label
$< \mathbf{1}\boldsymbol{\sigma}$	Normal
$\mathbf{1}\boldsymbol{\sigma} < \boldsymbol{value} < \mathbf{2}\boldsymbol{\sigma}$	Low/High
$> \mathbf{2}\boldsymbol{\sigma}$	Very Low/Very High

To evaluate the performance of our model, we consider the root of mean square error (RSME) and the mean absolute error (MAE) as the error performance measure.
MSE is computed as follow:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$$

MAE is calculated as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|$$

Where y_i, $\hat{y}_i$ are the actual value and predicted value respectively, N is the number of items in the train-validate dataset.

RESULTS AND DISCUSSION

The developed and trained model was used to make predictions on a partial selection of the overall dataset, allowing the predictions to be compared to the known values from the data. Random heats would be picked and provided to the software as inputs in a way matching the production environment. The outputs would then be analyzed and compared to the known results from the dataset. The resulting model shows good correlation with the known data, having an RMSE of 3.73°F for the midpoint temperature and 3.42°F for the LMF temperature loss. The MAE was for both was found to be less than three degrees Fahrenheit as well. The thermocouple accuracy for the data source is +/- 3°F, making the current prediction accuracy satisfactory.

Table 5. Error measurements on the training-validation dataset.

Measurement	Temperature at mid-point (° F)	LMF to tundish open temperature drop (° F)
MAE	2.89	2.61
RMSE	3.73	3.42

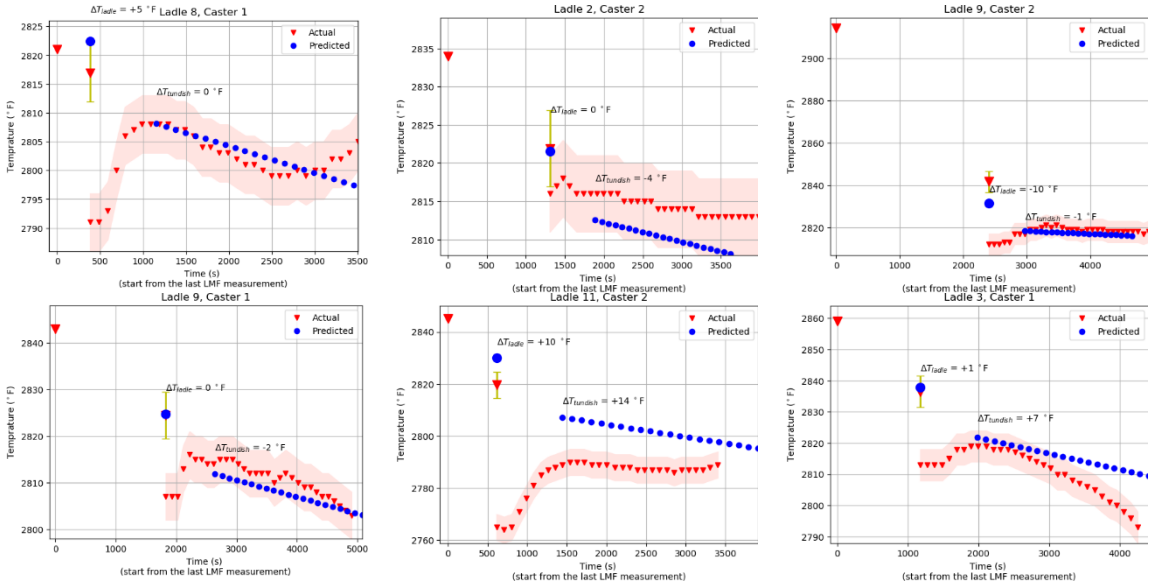

Figure 3. Plots of prediction values versus the actual data.

While the RSME and MAE show good performance, there still exist large outliers as seen in the histogram plots of Figure 4. While most cases performed within expected values, some heats show extreme errors of 10°F or more. Some of the larger errors are attributed to heats with abnormal conditions (excessively-large preheat or empty times, for example). The target accuracy for this software is to have few errors above 5°F, meaning more work is needed on identifying and processing abnormal cases. Additionally, work is being done to include other factors that are currently not part of the data processing procedure. These include LMF events such as arcing, stirring, and alloying.

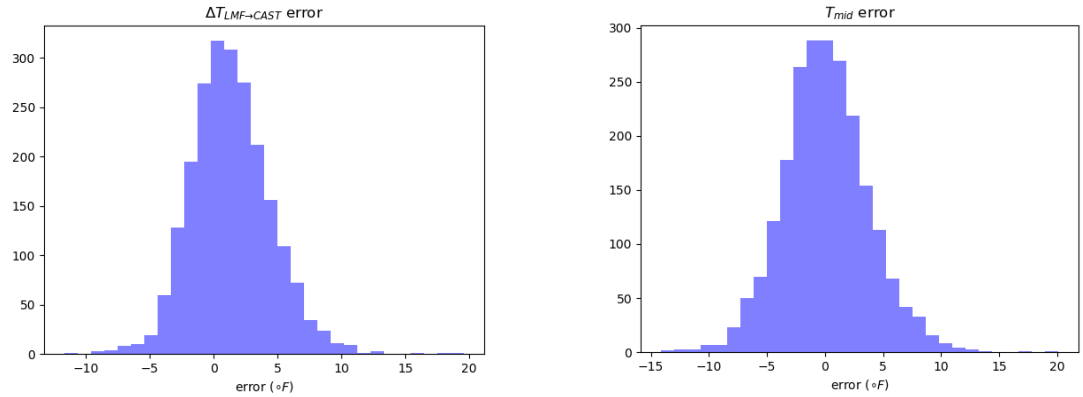

Figure 4. Error histograms for ladle temperature drop prediction (left) and tundish midpoint temperature prediction (right).

CONCLUSION

A deep learning software was developed for the purpose of providing ladle furnace operators predictions of ladle and tundish temperatures. The current work focuses on providing information on the heat loss in the ladle between the LMF and the caster as well as information on the tundish temperature behavior for that future heat. The model was developed in Python using a set of heat data provided by SDI Butler Division. The developed model has a mean absolute error and RMSE close to the accuracy of the thermocouples used to take temperature measurements, though extreme outliers exist still. Further model robustness is being developed to include additional data as well as enable the model to handle datasets from other production facilities. Finally, work on implementation in LMF operator displays is underway to enable live testing of the model.

ACKNOWLEDGEMENTS

The authors would like to thank AIST for providing funding through the *2019 AIST Digital Transformation Technologies* Grant, as well as for the support of SDI Butler Division, the Center for Innovation through Visualization and Simulation (CIVS) at Purdue University Northwest, and finally the members of the Steel Manufacturing and Visualization Consortium (SMSVC).

REFERENCES

[1] N. Naujok and H. Stamm, "Industry 4.0 in Steel: Status, Strategy, Roadmap and Capabilities Keynote Presentation Future Steel Forum, Warsaw," Warsaw, Poland, 2017.

[2] I. M. Cockburn, R. Henderson, and S. Stern, "The Impact of Artificial Intelligence on Innovation," in *NBER Conference on Research Issues in Artificial Intelligence*, 2017, no. September, pp. 1–39.

[3] J. M. Müller, D. Kiel, and K.-I. Voigt, "What Drives the Implementation of Industry 4 . 0 ? The Role of Opportunities and Challenges in the Context of Sustainability," *Sustainability*, vol. 10, no. 1, p. 247, 2018, doi: 10.3390/su10010247.

[4] H. Saxén and M. Sillanpää, "A model for decision support in continuous steel casting," *Model. Simul. Mater. Sci. Eng.*, vol. 2, no. 1, p. 79, 1994.

[5] T. P. Fredman, "Heat transfer in steelmaking ladle refractories and steel," *Scand. J. Metall.*, vol. 29, pp. 232–258, 2000.

[6] N. L. Samways and T. E. Dancy, "Factors affecting temperature drop between tapping and teeming," *J. Met.*, no. April, 1960.

[7] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 1, pp. 448–456, 2015.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers : Surpassing Human-Level Performance on ImageNet Classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[9] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Comput. Res. Repos.*, vol. 1506.01497, Jun. 2015, doi: 10.2307/j.ctt1d98bxx.10.