End-point Temperature Preset of Molten Steel in the Final Refining Unit Based on an Integration of Deep Neural Network and Multi-process Operation Simulation

Jianping YANG,¹) Jiangshan ZHANG,¹) Weida GUO,²) Shan GAO²) and Qing LIU¹,³)*

1) State Key Laboratory of Advanced Metallurgy, University of Science and Technology Beijing, Beijing, 100083 China.
2) Laiwu Iron and Steel Group Yinshan Section Steel Co., Ltd., Jinan, 271104 Shandong, China.

(Received on September 7, 2020; accepted on April 13, 2021; J-STAGE Advance published date: May 27, 2021)

End-point temperature preset of molten steel in the final refining unit is as important as its prediction for casting temperature control. However, it has not been given sufficient concern yet, and the proposed preset models in the literature usually cannot be used as practical tools due to their inherent shortcomings, e.g., oversimplifications made to a real environment during modelling. In this study, a novel preset approach was developed by integrating deep neural network (DNN) and multi-process operation simulation (MOS). By using MOS, the accurate transfer times of heats between the final refining unit and continuous caster can be solved before their actual scheduling, which is very significant for availability of the preset model based on DNN in practice. The DNN preset model was trained and tested with varying the values of hyper-parameters based on vast data points collected from a real steelmaking plant. Furthermore, preset models based on extreme learning machine (ELM) and multivariate polynomial regression (MVPR) were also established for comparison. The testing results indicate the DNN preset model with 3 hidden layers which contain 8, 4 and 2 neurons in sequence shows an advantage over other alternatives because of its evident improvement in preset accuracy and robustness. Meanwhile, a fine classification of data points considering metallurgical expertise can improve the generalization performance of the DNN preset model. The integrated approach has been applying in the studied steelmaking plant, and the ratio of qualified heats increases by 9.5% than before using it.

KEY WORDS: final refining unit; end-point temperature preset; molten steel; deep neural network; multi-process operation simulation; integrated approach.

1. Introduction

Due to the characteristics of high-temperature (1 500–1 700°C), multi-process, quasi-continuation along with various complex physical and chemical reactions among gas-liquid-solid multiphase, steelmaking-continuous casting process (SCCP) is still perceived as a crucial section in the whole steel manufacturing process.¹) Temperature control of molten steel, especially its casting temperature, is increasingly important for the final steel quality and energy consumption in SCCP.²–⁵) A poor casting temperature of molten steel can result in various quality problems of continuous casting strands, e.g. center segregation and surface crack, and even casting failure.

With increasingly stringent requirement for high-quality steel products, the temperature control of molten steel has been the focus of various modelling attempts, most of which are related to the end-point temperature prediction of molten steel in refining units. It is workable to control the casting temperature of molten steel effectively by predicting its end-point temperature in refining units, especially in the final one before casting. As reported in the literature, common methods for the end-point temperature prediction of molten steel in refining units include theoretical models based on heat transfer analysis,⁶) data models based on machine learning (ML)⁷) and integrated models combining them.⁸) With the fast development of artificial intelligence (AI), the ML models are more in favour owing to their powerful availability to deal with complex industrial problems.⁹,10) Another truth observed from previous researches is that the end-point temperature prediction of molten steel in ladle refining furnace (LF) has gained momentum as a

* Corresponding author: E-mail: qliu@ustb.edu.cn

© 2021 The Iron and Steel Institute of Japan. This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs license (https://creativecommons.org/licenses/by-nc-nd/4.0/).
result of its wide application in steelmaking process. Nath et al.11) developed an online model to predict the end-point temperature of molten steel in LF unit by analyzing the balance relations of materials and heat synthetically. Feng et al.12) proposed a combined method of case-based reasoning (CBR) and Bayesian Belief Network (BBN) to predict the end-point temperature of molten steel in LF unit. Fu et al.13) established a gray-box model to achieve the end-point temperature prediction of molten steel in LF unit, which was integrated by a heat transfer model and a BP neural network model. Apart from LF, the end-point temperature prediction in other refining units were also carried out. e.g., Ruhrstahl Heraeus Vacuum Degasser (RH), Argon Oxygen Decarburization (AOD) and Vacuum Oxygen Decarburization (VOD).14-16) Nowadays, the actual casting temperature of molten steel frequently deviates from the expected one, which is shown as Eq. (1):

\[
\theta = \theta - \eta \nabla \theta J(\theta) - \eta \lambda \nabla \theta \Omega(\theta)
\]  

(1)

Where, \(\theta\) indicates the collection of \(W^{(h)}\) and \(b^{(h)}\) in DNNs. \(J(\theta)\) is a loss function used to evaluate the error cost of DNN after each iteration, and mean square error (MSE) and root mean square error (RMSE) are widely used, displayed as Eqs. 2 and 3, where \(n\), \(Y_{\text{tar}}\) and \(Y_{\text{cal}}\) are the total number of data points in training or testing datasets, the target (actual) and calculated values of output variable, respectively. \(\Omega(\theta)\) in Eq. (1) is the regularizer used to reduce the risk of DNN overfitting. The lasso23) and weight decay24) are broadly used as regularization methods. \(\eta\) and \(\lambda\) represent learning rate and regularization coefficient, respectively.

\[
\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (Y_{\text{tar}} - Y_{\text{cal}})^2
\]  

(2)

\[
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (Y_{\text{tar}} - Y_{\text{cal}})^2}
\]  

(3)

In fact, the DNN with multiple hidden layers may not achieve a good generalization performance when only being trained by BP algorithm since it is prone to get stuck into poor local solutions, and thus a promising strategy has been proposed by combing pre-training and fine-tuning.20) In the process of pre-training, a hidden layer would be trained when its previous one has been trained completely instead of all hidden layers being trained synchronously. Restricted Boltzmann machine (RBM)25) and stacked auto-encoder (SAE)26) are extensively used as the pre-training approaches. After pre-training, the fine-tuning is carried out to search the global optimal weights using BP algorithm. In addition to reaching a better generalization performance, the computation cost for pre-training and fine-tuning is also lower than that for training only by BP algorithm.

By analyzing on previous researches related to the application of DNN,19,21,27,28) the hidden layer number as well as the neuron number per hidden layer have great influences on the generalization performance of DNN. With respect to other hyper-parameters in DNN, e.g., learning rate and iteration number, Refs. 19,21,27,28) also suggested some

---

2. Overviews of DNN and MOS

2.1. DNN

Artificial neural network (ANN) is composed of input layer, hidden layer and output layer, and the ones with no less than 3 hidden layers are generally deemed as DNN. Owing to the advantage in learning the representations of complex problems with high-dimensional data space, DNN and its derivatives such as convolutional neural network (CNN) have shown state-of-the-art performance in voice/pattern recognition and many industrial applications such as material engineering.18-20) Figure 1 shows a schematic diagram of DNN with K hidden layers and the workflow for learning as a package of mathematical formulas. The introduction of activation function makes it possible for the neural network to conduct a non-linear mapping learning. In particular, the rectified linear units (ReLU) \(f(z) = \max(0, z)\) has gradually become a promising activation function for DNN because it can efficiently avoid a gradient vanishing problem when using the sigmoid function \(f(z) = \frac{1}{1 + e^{-z}}\) as the activation function.21,22)

In general, the initial \(W^{(1)}\) and \(b^{(1)}\) in DNNs are randomly generated, and so the DNNs should be trained iteratively to find out the optimal ones with a limited iteration number or error cost. At present, the backpropagation (BP) algorithm based on gradient descent is a more prevalent learning method, and its mathematical expression for one iteration is shown as Eq. (1):

\[
\theta = \theta - \eta \nabla J(\theta) - \eta \lambda \nabla \Omega(\theta)
\]  

(1)

Where, \(\theta\) indicates the collection of \(W^{(h)}\) and \(b^{(h)}\). \(J(\theta)\) is a loss function used to evaluate the error cost of DNN after each iteration, and mean square error (MSE) and root mean square error (RMSE) are widely used, displayed as Eqs. 2 and 3, where \(n\), \(Y_{\text{tar}}\) and \(Y_{\text{cal}}\) are the total number of data points in training or testing datasets, the target (actual) and calculated values of output variable, respectively. \(\Omega(\theta)\) in Eq. (1) is the regularizer used to reduce the risk of DNN overfitting. The lasso23) and weight decay24) are broadly used as regularization methods. \(\eta\) and \(\lambda\) represent learning rate and regularization coefficient, respectively.
empirical values as reference.

2.2. MOS

To realize intelligent manufacturing in SCCP, the optimization of multi-process operation (mainly denoted as production scheduling) is as significant as the improvement of single-process techniques. However, the performance of most scheduling models established using traditional methods, such as the mathematical programming, cannot be satisfactory in a real environment because of less consideration of crane running and ladle cycling. Owing to a great power to describe complex industrial systems, the simulation method based on Plant Simulation software has been gradually utilized to study the multi-process operation problem in SCCP. He et al. applied Plant Simulation to understand the influence of workshop layout and process parameters on operation efficiency of SCCP. Deng et al. uncovered the effect of operation modes of duplex converter on equipment efficiency and steel output of SCCP using Plant Simulation.

Based on Plant Simulation, a MOS model was developed by our research group to simulate the multi-process operation taking crane running and ladle cycling into account. Figure 2 shows the visualization of a simplified MOS model containing one converter (BOF), one ladle refining furnace (LF) and one continuous caster (CCM). In Fig. 2, the locations of each processing unit are marked by red lines across the “Track”, and they also enable to sense the real-time positions of two cranes. The indications of main modelling objects in the simplified MOS model are also explained in the bottom half of Fig. 2. With the constraint of multi-heat continuous casting in CCM, a strategy of heat scheduling being reversed was employed in the MOS model. Equations (4) and (5) reveal how the heat scheduling is reversed in CCM, where the cast number is assumed as \( m \), and ending cast times of the \( m \) casts are presented as \( t_1 < t_2 < \ldots < t_i \ldots < t_m \).

$$
\begin{align*}
Y_i^I &= f(\sum_j W^{(L)}_{ij} X_j + b_{yi}^I) \\
Y_i^{(k)} &= f(\sum_j W^{(k)}_{ij} Y_j^{(k-1)} + b_{yi}^{(k)}), \quad k = 2, 3, \ldots, K \\
Y_i &= f(\sum_j W^{(K+1)}_{ij} Y_j^{(K)} + b_{yi}^{(K+1)})
\end{align*}
$$

A linear summation of the weighted inputs

Where, \( t_i^* \) and \( t_{ij}^* \) are starting cast times of cast \( i \) and heat \( j \) in cast \( i \), respectively, being represented in simulation; \( t \) is the casting cycle of CCM, and \( N_i \) is the heat number of cast \( i \). By using the above strategy, the simulated scheduling solved by the MOS model is an inverse duplication of the actual one. In Fig. 2, for instance, the tapping of molten steel in BOF is prior to oxygen blowing. Hence, the operations of processing units in the simulation cannot be equivalent.

![A schematic diagram of DNN with K hidden layers and the workflow for learning. (Online version in color.)](image-url)
to their real ones.

Figure 3 presents a detailed workflow of the MOS model, including the preparation of an initial scheduling solution as input of the MOS model, the simulation considering crane running and ladle cycling, and the output of the final simulated scheduling solution. In Fig. 3, the rules for crane running and ladle cycling as addressed in Step 2 are the key to hold the simulation closely representing the real environment. Through on-site survey, not limited to the studied steelmaking plant, the movement of each crane in tapping (refining) span is allowed in a specified area to avoid frequent conflictions between adjacent cranes. However, the conflictions between them are hard to prevent under some complex scenarios. In such cases, a strategy to eliminate the conflictions is employed in terms of the priority of missions loaded by two cranes, which is determined by the mission scheduling. For ladle cycling, the main work is to select a ladle from multiple candidates for online cycling, and it is reasonable that the ladle with the longest waiting time in “LadleSta” has the highest priority for online cycling. Since detailed formulaic descriptions on algorithms for crane running and ladle cycling have been reported in Ref. 31, this paper does not repeat the illustrations. After simulation as shown in Step 3, the heat scheduling must be reversed again referring to Eqs. (4) and (5), and then a usable scheduling solution is obtained.

By sufficient testing experiments in Ref. 31 for the same steelmaking plant studied in this paper, the availability of the MOS model has been proved, and for almost all tested heats, their mean error in transfer time was about 10.8% that is considerably acceptable for a harsh steelmaking environment. Therefore, the MOS model based on Plant Simulation is a reliable candidate to study the multi-process operation problem in SCCP, and furthermore it is more reasonable to adopt the simulated scheduling solution for practical use rather than the initial one.
3. Integration of DNN and MOS

3.1. Determination of Input Variables

To develop a DNN model for presetting the end-point temperature of molten steel in the final refining unit, the first step is to determine input variables. As reported in Ref. 17, the casting temperature of molten steel was influenced by following factors: end-point temperature of molten steel in the final refining unit, molten steel mass before casting, transfer time of the corresponding heat from the final refining unit to continuous caster and ladle thermal state. Therefore, it is believable to preset the end-point temperature of molten steel in the final refining unit by using the casting temperature, the molten steel mass before casting, the transfer time of the corresponding heat between the final refining unit and continuous caster as well as the ladle thermal state. As reported in Refs. 17,33,34), the temperature and cooling time of ladle lining before tapping and its thickness (erosion degree) had noticeable impacts on the heat transfer behavior of ladles. So, quantizing the ladle thermal state is workable by the above variables. However, the lining thickness is hard to measure directly by hardware tools in hand, and thus empty ladle mass is an alternative to it. Furthermore, the data on lining cooling time before tapping has not been recorded accurately in the studied steelmaking plant, and thus, the data on lining temperature before tapping is the alternative to it based on a close relationship between the lining temperature and the lining cooling time, since a longer lining cooling time means more heat loss and further a lower lining temperature. From the above, Table 1 lists the potential input variables of the DNN preset model. Using 2,250 data points offered by the studied steelmaking plant, the correlations between the input and output variables were validated by Pearson correlation coefficient and its significance testing based on the Student’s t-Test. The calculation methods of Pearson correlation coefficient and its significance are shown as Eqs. (6) and (7).

\[
\begin{align*}
    r & = \frac{\sum_{i=1}^{n} (X_{pi} - \bar{X}_p)(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n} (X_{pi} - \bar{X}_p)^2 \sum_{i=1}^{n} (Y_i - \bar{Y})^2}} \quad \cdots (6) \\
    t & = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \quad \cdots (7)
\end{align*}
\]

Table 1. The results of the correlations between the input and output variables.

<table>
<thead>
<tr>
<th>Potential input variables</th>
<th>Correlation coefficient</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Molten steel mass before casting/t</td>
<td>-0.231</td>
<td>0.000</td>
</tr>
<tr>
<td>Empty ladle mass/t</td>
<td>0.012</td>
<td>0.202</td>
</tr>
<tr>
<td>Lining temperature before tapping/°C</td>
<td>0.340</td>
<td>0.000</td>
</tr>
<tr>
<td>Transfer time of the corresponding heat between the final refining unit and continuous caster/min</td>
<td>-0.351</td>
<td>0.000</td>
</tr>
<tr>
<td>Casting temperature/°C</td>
<td>0.907</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Where, \( n \) is the total number of data points; \( X_{pi} \) and \( Y_i \) are the \( p \)-th input variable and output variable for all data points, respectively; \( \bar{X}_p \) and \( \bar{Y} \) are the mean values of the \( p \)-th input variable and output variable for all data points, respectively. Herein, the data from the first heats in each cast was not collected into the set of 2,250 data points because of their specific temperature schedule with a compensation to temperature loss of the tundish. Meanwhile, for each heat, the mean value of casting temperatures measured in three time points was used to calculate the correlation with the output variable and train the studied models. The results of the correlations between the input and output variables are given in Table 1.

The significances of the correlation coefficients are usually characterized by \( P \)-value. The lower \( P \)-value indicates a more significant correlation between the input and output variables. In particular, the correlation is extremely significant if the \( P \)-value is less than 0.01. It is found from Table 1 that most potential input variables show significant correlations with the output variable except for the empty ladle mass. Hence, the final input variables for DNN modeling include the casting temperature, the lining temperature before tapping, the transfer time of the corresponding heat between the final refining unit and continuous caster as well as the molten steel mass before casting.

3.2. Architecture and Workflow of the Integrated Approach

After determining the input variables, it is observed that the DNN preset model cannot be directly applied in actual production as one of the input variables, i.e. the transfer time of the corresponding heat between the final refining unit and continuous caster, is difficult to know accurately before its actual scheduling. In view of this, a practical approach based on the integration of DNN and MOS was developed for the studied steelmaking plant. Figure 4 describes the architecture and workflow of the integrated approach. Evidently, the DNN preset model is the main body of the integrated approach, while the MOS model calculates the transfer times of each heat in plans between the final refining unit and continuous caster and feeds them to the DNN preset model. As shown in Fig. 4, an initial scheduling for \( N \) heats is solved in advance by a scheduling model that was fully described in Ref. 35). Then the MOS model produces a simulated scheduling within several minutes once it receives the initial one. In other words, the transfer times of \( N \) heats between the final refining unit and continuous caster can be known before their actual scheduling and fed to the DNN preset model in sequence according to which heat being processed. With respect to other input variables, their values can be directly measured online except for the casting temperature. Figure 5 shows time stamps to indicate when the DNN preset model receives the values of other input variables. In most refining processes, alloy composition trimming is commonly the last charging operation, when the DNN preset model receives the initial one. In other words, the transfer times of \( N \) heats between the final refining unit and continuous caster can be known before their actual scheduling and fed to the DNN preset model in sequence according to which heat being processed.
temperatures for each steel grade are known in advance. However, the actual casting temperatures are often far from the expected ones. In this study, the actual casting temperatures were utilized to train and test the DNN preset model, whereas the expected ones were used for real preset of the end-point temperature of molten steel in the final refining unit. Based on plenty of production experience in Q345B steel, a major product in the studied steelmaking plant, the suitable range for its casting temperature is from 20 to 30°C (superheat degree) over the liquidus temperature. Thus, the expected casting temperature for Q345B steel is set as 25°C over the liquidus temperature, just as the study in Ref. 36).

Once receiving the values of all input variables corresponding to heat \( j \), the DNN preset model starts to run and then releases a preset value to the final refining unit within a few seconds. Hereby, the operators can receive the preset value before soft blowing as shown in Fig. 5. For the first heats in each cast, the preset values for them are obtained by adding an empirical compensation value of 20°C to the initial preset ones. Then a decision can be made on the subsequent schedule in heating and soft blowing in a given time through a comparison between current actual temperature of molten steel and its preset value. With respect to the mentioned given time, it is determined based on a compromise between the simulated and actual scheduling. It is, however, inevitable that the actual scheduling deviates from the simulated one. Hence, the initial and simulated scheduling must be adjusted in accordance with the actual starting cast times of each heat.

3.3. Setting of the DNN Preset Model

Indeed, the performance of the integrated approach is decided by the MOS and DNN preset models. Ref. 31) has verified the availability of the MOS model, and so it is a crucial work to establish a high-efficient DNN preset model. In particular, the settings of hyper-parameters are very important for usability of the DNN preset model. As representation in Section 2.1, the numbers of hidden layers and neurons in each hidden layer have great effects on the generalization performance of DNN. As reported in Refs. 19,21,27,28), the DNN models with hidden layers of 2 to 4 were suitable to solve the regression problems with low-medium dimensional parameter space. Moreover, the DNN model proposed in Ref. 27) exhibited an optimal generaliza-
tion performance when \( \varphi \) (the ratio of the neuron number in the first hidden layer to the input variable number) and \( \lambda \) (the ratio of the neuron numbers in the next and previous hidden layers) were equal to 2 and 0.5, respectively. Models 2, 4 and 6 shown in Table 2 were established based on the study in Ref. 27, and other models shown in Table 2 were established to make comparison by setting \( \varphi \) as 1, 3 and 4. In addition, the settings of other main hyper-parameters for 8 preset models are summarized in Table 3, which were utilized to find the optimal hidden layer number and neuron number per hidden layer. After that, learning rate, regularization coefficient and iteration number were further optimized through additional training. As suggested in Ref. 37, Models 1–5 with 1 or 2 hidden layers were trained only by BP algorithm, and Models 6–8 with more than 2 hidden layers were trained by combining pre-training based on SAE and fine-tuning.

From 2,250 data points offered by the studied steelmaking plant, 450 data points were randomly extracted as a testing dataset, and the remaining 1,800 data points were treated as a training dataset. Before modelling, the original values of the input and output variables were normalized to between 0 and 1 in the same dimensionless unit to prevent the occurrence of low preset accuracy resulted from the disparity on the input and output variables. After normalization, the training dataset and the remaining 1,800 data points were treated as testing dataset. Before modelling, the original values of the input and output variables were normalized to between 0 and 1 in the same dimensionless unit to prevent the occurrence of low preset accuracy resulted from the disparity on the input and output variables. After that, learning rate, regularization coefficient and iteration number were further optimized through additional training. As suggested in Ref. 37, Models 1–5 with 1 or 2 hidden layers were trained only by BP algorithm, and Models 6–8 with more than 2 hidden layers were trained by combining pre-training based on SAE and fine-tuning.

From 2,250 data points offered by the studied steelmaking plant, 450 data points were randomly extracted as a testing dataset, and the remaining 1,800 data points were treated as a training dataset. Before modelling, the original values of the input and output variables were normalized to between 0 and 1 in the same dimensionless unit to prevent the occurrence of low preset accuracy resulted from the disparity on the input and output variables. After that, learning rate, regularization coefficient and iteration number were further optimized through additional training. As suggested in Ref. 37, Models 1–5 with 1 or 2 hidden layers were trained only by BP algorithm, and Models 6–8 with more than 2 hidden layers were trained by combining pre-training based on SAE and fine-tuning.

3.4. Evaluation of the DNN Preset Model

In this study, the hit ratios within error of ±3 and ±5°C, RMSE and the goodness of fit \( R^2 \) were taken into account as performance metrics of the DNN preset model. The mathematical expression of \( R^2 \) is displayed as Eq. (8), where \( \overline{Y}_{\text{tar}} \) indicates the mean actual value of output variable for all data points in the training or testing datasets, and the indications of other parameters are the same as the corresponding ones in Eq. (3). It is well known that the larger \( R^2 \) and the smaller RMSE demonstrate the better performance of models. Since the preset accuracy for the testing dataset directly indicates the generalization performance of the DNN preset model in practice, the experimental results related to the testing dataset were recorded to make comparison.

\[
R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_{\text{tar}} - Y_{\text{pred}})^2}{\sum_{i=1}^{n}(\overline{Y}_{\text{tar}} - Y_{\text{tar}})^2} \tag{8}
\]

4. Results and Discussion

4.1. Effect of Hyper-parameters on Model Performance

In this study, the programs for 8 preset models were written using Python language. Each of the preset models was trained and tested 50 times on a personal computer with Intel (R) i5-3770U CPU/8.0 GB RAM/Window 10, and the training and testing datasets were re-determined before each of the trials. Table 4 shows the testing results of 8 preset models, each of which is the mean value for 50 trials. The scatter plots of the hit ratios within error of ±3 and ±5°C for testing each preset model 50 times are drawn in Figs. 6(a) and 6(b). Furthermore, the mean computation costs for training each preset model 50 times are also listed in Table 4 to estimate their potential update rates in practice. However, there is no an additional discussion on the computation costs for testing each preset model, which are less than a second. It is not hard to observe from Table 4 that the best results of the hit ratios within error of ±3 and ±5°C, RMSE and \( R^2 \) are 76.8%, 93.6%, 3.1868 and 0.9274 respectively, which are all addressed in the same row corresponding to Model 6. Oppositely, the worst results of above evaluation indicators are found in the same row corresponding to Model 1. In Figs. 6(a) and 6(b), the differences between the maximum and minimum values of the hit ratios within error of ±3 and ±5°C for Model 6 are 6% and 5.3%, respectively. Indeed, the scattered degrees of hit ratios corresponding to Model 6 are lower than those corresponding to other models in the cases of error within either ±3 or ±5°C. That means

<table>
<thead>
<tr>
<th>Table 2.</th>
<th>Network structures of 8 preset models.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>Hidden layer number</td>
</tr>
<tr>
<td>Model 1</td>
<td>1</td>
</tr>
<tr>
<td>Model 2</td>
<td>1</td>
</tr>
<tr>
<td>Model 3</td>
<td>2</td>
</tr>
<tr>
<td>Model 4</td>
<td>2</td>
</tr>
<tr>
<td>Model 5</td>
<td>2</td>
</tr>
<tr>
<td>Model 6</td>
<td>3</td>
</tr>
<tr>
<td>Model 7</td>
<td>3</td>
</tr>
<tr>
<td>Model 8</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3.</th>
<th>Settings of other main hyper-parameters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>Values</td>
</tr>
<tr>
<td>Learning method</td>
<td>BP(pre-training + fine-tuning)</td>
</tr>
<tr>
<td>Activation function in hidden layer</td>
<td>ReLU</td>
</tr>
<tr>
<td>Activation function in output layer</td>
<td>Linear ((y = x))</td>
</tr>
<tr>
<td>Loss function</td>
<td>RMSE</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.01</td>
</tr>
<tr>
<td>Regularization coefficient</td>
<td>0.001</td>
</tr>
<tr>
<td>Batch size</td>
<td>20</td>
</tr>
<tr>
<td>Iteration number</td>
<td>20 000</td>
</tr>
</tbody>
</table>

| Table 4. | Testing results of 8 preset models. |
Model 6 shows a stronger robustness. In addition to Model 6, Models 7 and 8 also show acceptable testing results, and however, the computation costs for training them are higher than that for training Model 6 because of more neurons or hidden layers in them. It is true that the lower computation cost for training Model 6 indicates a faster update rate for its practical use with the accumulation of historical data points. In summary, Model 6 is a more promising candidate to preset the end-point temperature of molten steel in the final refining unit based on a comprehensive consideration to preset the end-point temperature of molten steel in the

4.2. Effect of Data Attributes on Model Performance

The studied steelmaking plant has four CCMs to produce multiple grades of steel products with different process routes, in which the physical and chemical properties of molten steel are different. Meanwhile, there is also a difference in ladle thermal state between ladles that cycle around SCCP continuously and the reused ones after maintenance. Hence, it may be not appropriate that using data points from the production of common steels trains and tests the DNN preset model for high-quality steels. Thus, 2,250 data points were classified into different datasets based on the steel grade and ladle type to further explore the effect of data attributes on model performance. The detailed classifications discussed in this section are listed in Table 5.

As the testing results in Section 4.1, Model 6 with the optimized hyper-parameters was chosen to conduct the study in this section. To validate its advantage as a kind of DNN model, ELM and MVPR were also applied to establish preset models and make comparison. Some main hyper-parameters of the ELM and MVPR preset models were determined based on plenty of trials, and the best ones are shown in Table 6. The indications of hyper-parameters are similar as the corresponding ones in the DNN preset model. To further understand the working mechanisms of ELM and MVPR, Refs. 39) and 40) are suggested for the interested readers.

The same evaluation indicators as the ones used in Section 4.1 were employed to compare the performances of the DNN, ELM and MVPR preset models under C-1, C-2 and C-3, respectively. Running each case 20 times, and the mean values of each evaluation indicator were recorded as shown in Table 7. The DNN preset model (Model 6) demonstrates noticeable improvements in almost all of the evaluation indicators under C-1, C-2 and C-3 against the correspond-
ing ones under the unclassified dataset as shown in Fig. 7(d). Taking the hit ratios within error of ±3 and ±5°C of Model 6 under C-3 as examples, there are increments of 3.5% and 3.4% respectively against the ones under the unclassified dataset. Furthermore, all evaluation indicators under C-3 are superior to the corresponding ones under C-1 and C-2. The same observation is obtained from the testing results of the ELM and MVPR preset models. Therefore, it is concluded that better performances of the preset models can be obtained when they are trained and tested by finer classification of data points. In addition, the results of evaluation indicators of the DNN preset model are all better than those of the ELM and MVPR preset models under C-1, C-2 and C-3 as shown in Table 7. The computation cost of 20 seconds for training the DNN preset model can be neglected in comparison to technological operations of tens of minutes in actual production. To sum up, the expected size of the training dataset may be chosen in the range from 800 to 1 800 for practical use of the DNN preset model, since there is a good trade-off between generalization performance and update rate.

4.3. Actual Application of the Integrated Approach

As discussed in Sections 4.1 and 4.2, Model 6 with the optimized hyper-parameters shows the best performance among multiple candidates to preset the end-point temperature of molten steel in the final refining unit. Thus, it

![Fig. 7](image)

Fig. 7. The values of RMSE with the increments of a) learning rate, b) regularization coefficient and c) iteration number as well as d) the values of each indicator under the optimized hyper-parameters. (Online version in color.)

<table>
<thead>
<tr>
<th>Item</th>
<th>Attribute</th>
<th>Size of training dataset/heat</th>
<th>Size of testing dataset/heat</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-1</td>
<td>Q345B a)</td>
<td>1 000</td>
<td>250</td>
</tr>
<tr>
<td>C-2</td>
<td>Ladle that cycles continuously b)</td>
<td>1 000</td>
<td>250</td>
</tr>
<tr>
<td>C-3</td>
<td>Both a) and b) c)</td>
<td>800</td>
<td>200</td>
</tr>
</tbody>
</table>

Table 5. Classifications of data points based on steel grade and ladle type.

Table 6. Hyper-parameters of the ELM and MVPR preset models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Hyper-parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELM</td>
<td>Neuron (node) number in hidden layer</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Regularization coefficient</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>Activation function</td>
<td>Sigmoid</td>
</tr>
<tr>
<td>MVPR</td>
<td>Dimension</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Regularization method</td>
<td>Bayesian bridge</td>
</tr>
<tr>
<td></td>
<td>Regularization coefficient</td>
<td>0.001</td>
</tr>
</tbody>
</table>
was used to develop the integrated approach with the MOS model.

Before applying the integrated approach in the studied steelmaking plant, the decision on the end-point temperature of molten steel in the final refining unit was made in the dependence of artificial expertise.

Table 7. Testing results of the DNN, ELM and MVPR preset models under C-1, C-2 and C-3.

<table>
<thead>
<tr>
<th>Classification</th>
<th>Model</th>
<th>Hit ratio within error of ±3°C/§</th>
<th>Hit ratio within error of ±5°C/§</th>
<th>RMSE</th>
<th>( R^2 )</th>
<th>Time for training/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN (Model 6)</td>
<td>76.2</td>
<td>94.4</td>
<td>3.1537</td>
<td>0.9290</td>
<td>26.82</td>
<td></td>
</tr>
<tr>
<td>C-1</td>
<td>ELM</td>
<td>72.9</td>
<td>91.7</td>
<td>3.3295</td>
<td>0.9211</td>
<td>1.27</td>
</tr>
<tr>
<td></td>
<td>MVPR</td>
<td>71.6</td>
<td>91.2</td>
<td>3.4064</td>
<td>0.9172</td>
<td>1.45</td>
</tr>
<tr>
<td>DNN (Model 6)</td>
<td>77.6</td>
<td>95.3</td>
<td>3.0928</td>
<td>0.9317</td>
<td>26.73</td>
<td></td>
</tr>
<tr>
<td>C-2</td>
<td>ELM</td>
<td>75.3</td>
<td>92.5</td>
<td>3.2361</td>
<td>0.9255</td>
<td>1.32</td>
</tr>
<tr>
<td></td>
<td>MVPR</td>
<td>76.0</td>
<td>93.2</td>
<td>3.1913</td>
<td>0.9271</td>
<td>1.50</td>
</tr>
<tr>
<td>DNN (Model 6)</td>
<td>80.5</td>
<td>97.0</td>
<td>2.9206</td>
<td>0.9396</td>
<td>22.28</td>
<td></td>
</tr>
<tr>
<td>C-3</td>
<td>ELM</td>
<td>78.5</td>
<td>95.5</td>
<td>3.0390</td>
<td>0.9348</td>
<td>1.03</td>
</tr>
<tr>
<td></td>
<td>MVPR</td>
<td>77.5</td>
<td>95.0</td>
<td>3.0962</td>
<td>0.9323</td>
<td>1.16</td>
</tr>
</tbody>
</table>

Fig. 8. Statistical results of superheat degree of molten steel corresponding to the heats produced a) before and b) after applying the integrated approach. (Online version in color.)

Table 8. Statistical results of three key indexes before and after applying the integrated approach.

<table>
<thead>
<tr>
<th>Production index</th>
<th>Before application</th>
<th>After application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index 1/ton</td>
<td>352.0</td>
<td>117.6</td>
</tr>
<tr>
<td>Index 2/time</td>
<td>72.3</td>
<td>30.1</td>
</tr>
<tr>
<td>Index 3/time</td>
<td>2.5</td>
<td>0.9</td>
</tr>
</tbody>
</table>

30°C for casting is merely 87.0%. Since September 2019, the integrated approach has been applying, and the ratio of qualified heats reaches up to 96.5% in the period from September 2019 to March 2020 as shown in Fig. 8(b). Table 8 shows the statistical results of three key indexes in the studied steelmaking plant before (from April to July 2019) and after (from September 2019 to March 2020) applying the integrated approach, including the unqualified strand amount (Index 1), the adjustment number of casting speed (Index 2) and the casting failure number (Index 3) averagely per month. Herein, the unqualified strands denote the ones resulted from poor temperature control, not including the head and tail strands in each cast. It is observed from Table 8 that above three key indexes reduce by 66.6%, 58.4% and 64.0% respectively after applying the integrated approach. Apparently, the production status of the studied steelmaking plant has a significant improvement. In conclusion, the integrated approach is very efficient to achieve a fine control of the casting temperature by conducting a reliable end-point temperature preset of molten steel in the final refining unit.

5. Conclusions

In this study, a novel approach has been proposed based on the integration of DNN and MOS. Its practical use to preset the end-point temperature of molten steel in the final refining unit directly depends on the availability of the DNN preset model. The MOS model offers the DNN preset model an accurate transfer time of each heat between the final refining unit and continuous caster as one of its input variables. Meanwhile, casting temperature, lining temperature before tapping and molten steel mass before casting are also selected as input variables. Hyper-parameters of the optimal DNN preset model in this study, i.e., the hidden layer number, the neuron number per hidden layer, learning
rate, regularization coefficient and iteration number are 3, 8/4/2, 0.05, 0.003 and 23000, respectively. It shows a higher preset accuracy than the ELM and MVPR preset models. Owing to a fine classification of data points, the generalization performance of the DNN preset model can be further improved. After applying the integrated approach in the studied steelmaking plant, the ratio of qualified heats with a superheat degree of 20 to 30°C for casting has increased from 87.0% to 96.5%. This is beneficial to achieve a fine continuous casting procedure with constant casting speeds and accelerate the intelligent production with a higher quality and output of required steel products.

In our next work, a high-efficient integration involving the backward preset and forward prediction will be undertaken with more emphasis on temperature control of molten steel in the whole SCCP, not limited to the casting temperature.

Acknowledgements

This study was supported by the National Natural Science Foundation of China (Nos. 50874014 and 51974023) and the Fundamental Research Funds for Central Universities (No. FRF-BR-17-029A).

REFERENCES