

Received November 30, 2019, accepted January 21, 2020, date of publication January 27, 2020, date of current version February 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2969498

Spatial and Sequential Deep Learning Approach for Predicting Temperature Distribution in a Steel-Making Continuous Casting Process

SOO YOUNG LEE¹, BAYU ADHI TAMA¹, CHANGYUN CHOI¹, JONG-YEON HWANG², JONGGEUN BANG², AND SEUNGCHUL LEE¹

¹Department of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea

²POSCO Process and Engineering Research Laboratory, Gwangyang 57807, South Korea

Corresponding author: Seungchul Lee (seunglee@postech.ac.kr)

This work was supported in part by the Research and Development project of POSCO under Grant 2019Y023, in part by the Technology Innovation Program of the Ministry of Trade, Industry, and Energy (MOTIE) under Grant 10080729, in part by the Korea Institute for Advancement of Technology (KIAT) Grant funded by the Korean Government (MOTIE) (the Competency Development Program for Industry Specialist) under Grant N0008691, and in part by the High-Potential Individuals Global Training Program of Institute for Information and Communications Technology Planning and Evaluation (IITP) under Grant 2019-0-01589.

ABSTRACT Continuous casting is the procedure of the successive casting for solidification of the steel, which contains several cooling processes along the caster to coagulate the molten steel. It is such a rule of thumb that strand surface quality and casting productivity is highly dependent on temperature control. A finite-difference method is one of estimating temperature distribution, yet it hinders the process control efficiently. Song, *et al.* suggest a multimodal deep learning approach for prediction of the temperature. However, sequential and transient phenomena of solidifying steel are not considered, which makes it difficult to estimate the sequential heat-transfer characteristics in the whole process of the steel concretion. Herein, a deep learning model is proposed to predict the temperature distribution by taking into account both transient and steady-state characteristics. The proposed model addresses both spatial and sequential information by incorporating a convolutional neural network (CNN) and a recurrent neural network (RNN). Our quantitative and qualitative results show considerable predictive performance improvement against baseline models. Furthermore, the proposed model is applicable in a real-world steel-making industry by providing real-time temperature prediction, whilst retaining a lower computational cost.

INDEX TERMS Continuous casting, recurrent neural networks, convolutional neural networks, deep learning, steel industries, temperature distribution predictions.

I. INTRODUCTION

In the steel industry, continuous casting process, also known as strand casting process, refers to a consecutive casting technique via injecting molten steel into the mold and spray region, while solidifying the steel with cooling processes along with the caster. It is located in the middle stage of entire manufacturing processes, making intermediate materials, e.g., blooms, billets, and slabs, before some additional treatments and final steel-making procedures are performed [1]. Owing to some outstanding characteristics of this process, e.g. massive productivity, good quality, and

cost-efficiency, it is widely utilized for manufacturing intermediate casting products [2], [3].

The continuous casting process is a sequential solidification procedure from molten steel in *liquidus* to an intermediate product in *solidus*. As visualized in Figure 1, a cooling process is typically categorized into two main operations, i.e. primary and secondary cooling. The liquid molten steel which is stored in *ladle* is directly poured through a vessel, e.g. *tundish* and then the molten steel from the vessel is vertically poured into an enclosed copper mold. The poured steel is affected by a water-cooled copper mold as a way of primary cooling. Liquid steel releases the heat to the water-cooled mold, starting to be solidified inside the mold. Following the primary cooling, a secondary cooling reaches out a spray

The associate editor coordinating the review of this manuscript and approving it for publication was Kemal Polat.

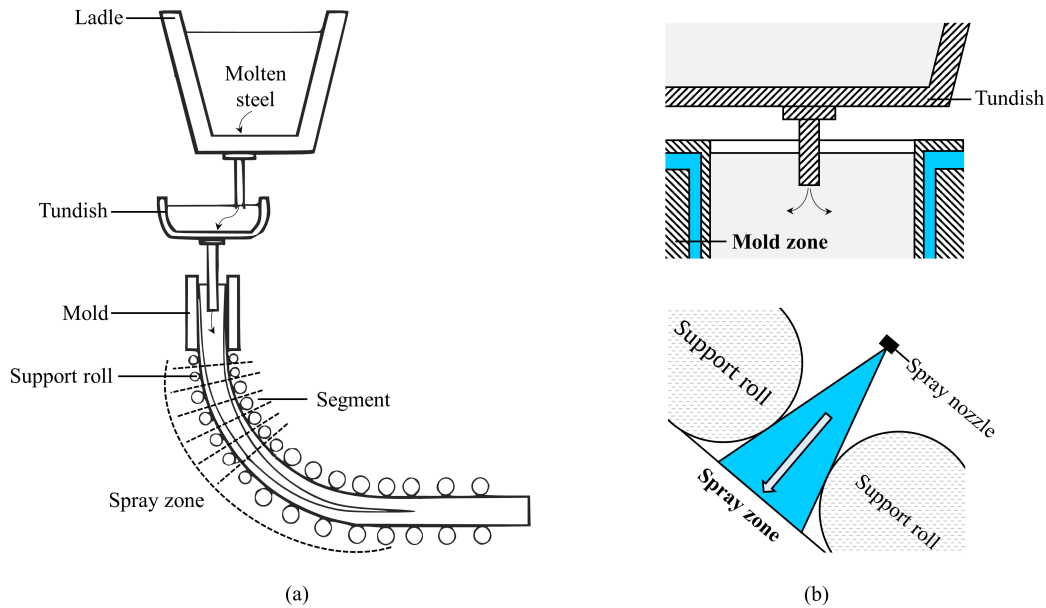


FIGURE 1. Schematic diagram of continuous casting process: (a) structure of continuous casting process (b) primary and secondary cooling in the mold and the spray region.

cooling zone, in which a variety of water spray nozzles and support rolls are lined up in a row. The spray zone includes several segments in the corresponding location to pair of rolls and nozzles, respectively. As the steel passes through the zone, a secondary cooling plays a critical role in extracting most of the total heat from the steel, influencing both surface and internal defects [4], [5]. The water flow rate is controlled by the valves of spray nozzles to achieve certain targeting patterns of the temperature [6]. Since the profile of the temperature is highly related to the quality and the productivity of the steel, accordingly, controlling the thermal activity of solidifying steel in the primary and secondary cooling process has been a significant issue in the steel industry.

The temperature of the steel greatly matters in a continuous casting process. Since the temperature is the result of sequential heat transfer from both cooling processes, the desired temperature could be properly obtained via manipulating such processes to ensure the quality of the product and safety. Several studies show that the temperature is closely related to the quality of the product. For instance, it is widely known that huge reheating of the steel has an effect on internal cracks and too low surface temperature affects surface and shape defects [7], [8]. Too high temperature is also able to cause the steel to detrimental bulging, even breakout that could put a lot of workers in jeopardy [9]. Consequently, understanding the inner and surface temperature distribution of the steel attributed to the behavior of heat transfer and solidification is necessary to make the casting process successful.

However, measuring the temperature of the steel in a real-world steel industry possesses several limitations. For instance, it is not possible to obtain a precise temperature distribution during the process owing to its variable

environment of phase transition and excessively high temperature over a thousand degrees Celsius according to its operational conditions. For this reason, indirect and sporadic methods only for detecting solidified shell thickness or a certain point of temperature are currently carried out [10]. The most typical method, for example, is called nail shooting which is destructive testing while the thickness is empirically analyzed by shooting a sulfur nail into the strand [11].

In an actual continuous casting process, a numerical analysis method of heat transfer is being currently utilized via a physics-based simulation model. The heat transfer model is developed to demonstrate thermal phenomena within the solidifying steel of the process mathematically [12], which gives profile information of the temperature. Figure 2 shows a flowchart of calculating the temperature and controlling the amount of water through spraying segments, which is dominantly regulated in a large number of continuous casting cases. First of all, the amount of cooling water in every segment is obtained by the sensors in spray nozzles. Based on several dependent variables, e.g. amount of cooling water and casting speed, the simulation model then calculates the temperature distribution of solidifying steel, followed by the comparison between the user-defined target temperature and the calculated one. Lastly, the PID controller recursively determines the proper amount of cooling water that satisfies the desired temperature.

The main drawback of such a conventional method is attributed to a model-based temperature calculation with finite-difference, whose computation resources are required excessively. Consequently, utilizing the simulation program for calculating the temperature is reasonably limited to only a small portion of the casting period. Moreover, it leads

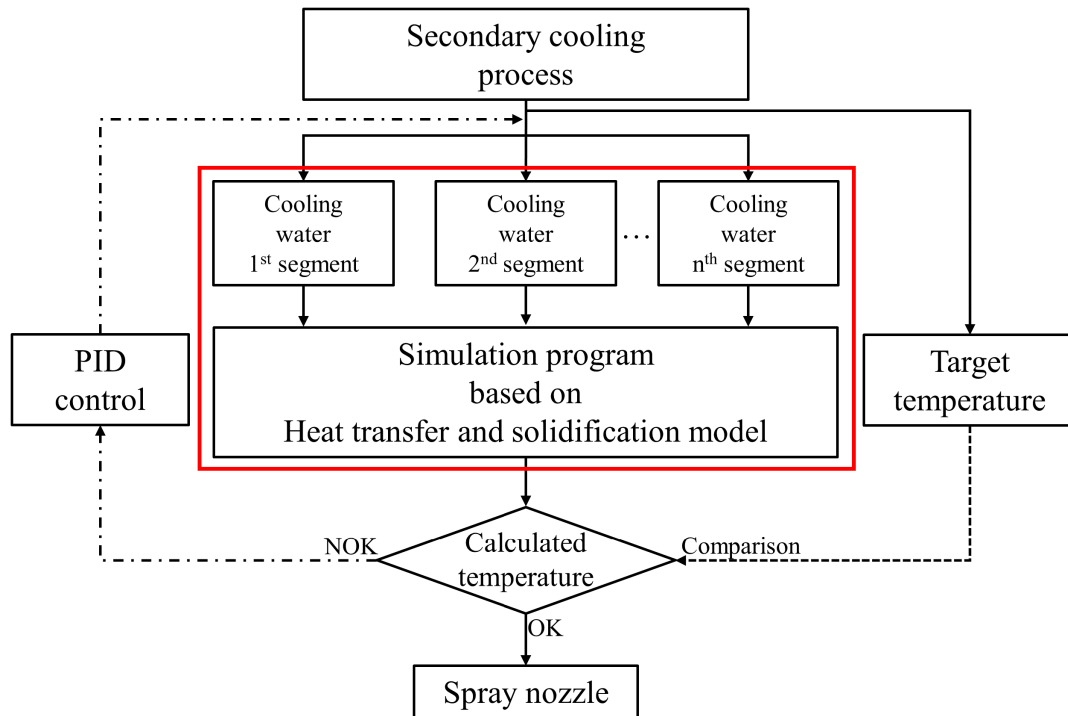


FIGURE 2. Common flow chart of calculating the temperature and controlling cooling water.

researchers to rely upon empirical lookup tables [6], hindering dynamic control of the process. Therefore, it is necessary to have a less-computation and prompt model at each time step for predicting the temperature within the solidifying steel and, by extension, for enhancing the capability of dynamic control.

Imbued by the aforementioned limitations, a real-time prediction model based on deep neural networks is developed for estimating the temperature distribution in a steel-making continuous casting process. A novel approach based on the mixture of convolutional neural network (CNN) and recurrent neural network (RNN) is dealt with in this paper, considering that the process data, e.g. casting speed and the amount of cooling water, are taken into account as a time series, whilst the temperature profile is considered as an image. We describe our proposed method using CNN and RNN in detail in Section III. The proposed method is then compared with existing methods in terms of predictive accuracy and computational cost.

The main contributions of the paper are as followings.

- Firstly, a new approach that combines two different deep neural network models, i.e. CNN and RNN for temperature prediction in a continuous casting process is proposed. It works based on the mechanism of encoder and decoder, in which input sequence data in numeric and image form are processed simultaneously.
- Secondly, prediction of both transient and steady-state temperature profile of the steel in every time step without any numerical analysis, based upon sequential

and spatial information extracted from end-to-end deep learning.

- Lastly, the development of reliable and real-time prediction model for on-site purpose with real-world data, based on the optimization and the assessment of the model.

We structure the remaining part of the paper as followings. Section II presents the background and preliminaries of the heat transfer model of continuous casting, as well as the overview of previous work. The proposed method and comparison strategies are described in Section III. Section IV describes the experimental results concerning predictive performance and computational time. Finally, Section V concludes the paper.

II. BACKGROUND AND PRELIMINARIES

In this section, the theoretical background of the heat transfer and solidification model in the continuous casting process used for data generation is concisely described, followed by an overview of neural network-based approaches in the steel-making industry.

A. HEAT TRANSFER MODEL OF SOLIDIFYING STEEL

A variety of thermal activities, i.e., heat conduction, radiation, and convection, occur in a continuous casting process, caused by the mechanical behavior of heat transfer and solidification. Under previous studies, mathematical models have been developed to describe the phenomena in primary and secondary cooling [12]–[14]. The most widely used approach

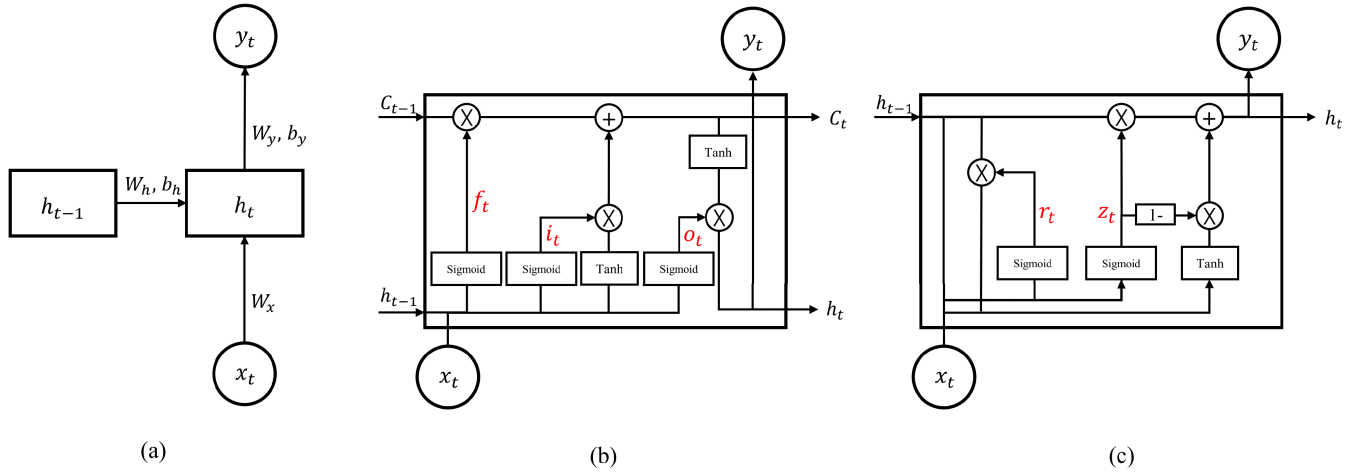


FIGURE 4. Schematic diagram of recurrent units: (a) basic recurrent neural network, (b) operation of LSTM unit and (c) operation of GRU unit.

with a convolutional neural network (CNN) and deep neural network (DNN) for optimizing temperature control in a continuous casting process.

We extend the work reported in [22] because it lacks describing sequential and transient phenomena of solidifying steel at each time step. The work solely utilizes the steady-state temperature distribution obtained from a simulation program, thus input conditions used for modeling, e.g. casting speed, the amount of cooling water and temperature distribution, are considered as constant, not as sequential values. Since the dynamic prediction of the temperature distribution is required for practical use, there are still some challenging problems, which are motivated in this study.

III. PROPOSED METHOD

In this section, the underlying theory about RNN and CNN are explained, followed by the proposed method, as well as a comparison procedure for evaluation.

A. FUNDAMENTAL OF RNN AND CNN

In this paper, a recurrent neural network (RNN) is utilized to predict both transient and steady-state temperature distribution at each time step, considering the connections across different time indices. RNN is designed to process sequential data, which has been successfully used in various kinds of tasks with time-series data, e.g., forecasting [23]–[25]. While each input is fed into the network independently in a traditional feedforward neural network, RNN utilizes the memory between hidden nodes across the sequence of the input. The hidden states are trained to represent the temporal information until the past to the present. Figure 4(a) shows a schematic diagram of RNN with basic recurrent units. The h_t , hidden state at time steps t , plays a role as a memory. It is a function of a previous hidden state h_{t-1} and an input vector at time step t , x_t . Mathematically, it could be expressed as follows:

$$h_t = f(h_{t-1}, x_t) = g(W_h h_{t-1} + W_x x_t + b_h) \quad (6)$$

where g is a nonlinear activation function, e.g., *sigmoid* and *tanh* function. Weights and biases are denoted as W and b , respectively.

Subsequently, long short-term memory (LSTM) unit and gated recurrent unit (GRU) are proposed to overcome the existing problem of basic RNN, i.e., a vanishing gradient that prevents the network from capturing long-term dependency. As shown in Figure 4(b), the LSTM unit in hidden layer has three different gates, i.e., forget gate f_t , input gate i_t and output gate o_t , to accumulate the information over a long duration, where the gates define whether the old states are forgettable or memorable [26]. To achieve this, cell state at time step t , c_t , is newly added to the unit, which is determined by f_t and i_t . The f_t controls the memory with *sigmoid* activation function based on input vector x_t and prior hidden state h_{t-1} , while i_t controls remembering current information. Finally, the hidden state h_t is calculated using o_t and c_t with *tanh* activation. Furthermore, the GRU unit has been proposed as a simplified version of the LSTM to overcome a large amount of learning parameters [27]. The hidden state at time step t , h_t is determined using an update gate z_t and a reset gate r_t , while no longer a cell state exists in the unit (see Figure 4(c)). The former controls how much information from the previous hidden state will carry over to the current one, whilst the latter decides whether the previous hidden state is forgettable [27].

In this study, a convolutional neural network (CNN) is utilized as a feature extractor of the temperature distribution image. CNN is the most efficient neural network algorithm for image processing and classification, which can automatically learn spatial information of the image [28]. As shown in Figure 5, it is commonly made up of several types of layers, i.e., convolutional, pooling and fully connected layer. First of all, a convolutional layer is used to extract the abstract representation from an image using consecutive convolution operations, while the feature maps are produced via the operations. It is followed by the pooling layer as a way of *downsampling* technique, e.g., max-pooling,

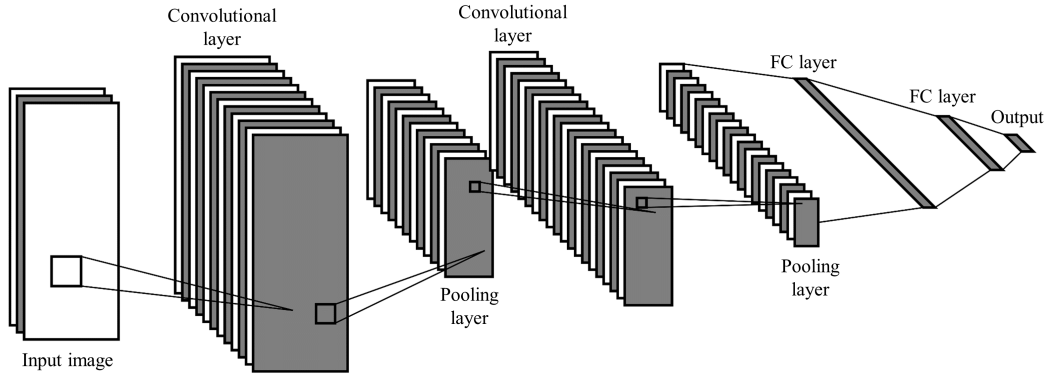


FIGURE 5. A typical architecture of convolutional neural network.

which finds maximum values from each sub-regions in the feature maps. A pooling layer is carried out to reduce the spatial resolution of the feature maps. It takes advantage of decreasing the number of trainable parameters, as well as avoiding *overfitting* of the model and emphasizing specific parts of the spatial characteristics. Lastly, the fully connected layer utilizes high-level features extracted and gathered by convolutional and pooling layers for classifying the input image that belongs to a particular class.

B. PROPOSED MODEL

The key part of this paper is a combination of two different neural network architectures, i.e. deep recurrent neural network (RNN) and deep convolutional neural network (CNN), for addressing continuous casting process sequential data in numeric and image form. As introduced previously, the RNN is designed to process the sequential numeric data with hidden states that represent temporal information, whilst CNN learns the spatial information of an image. Since our goal is to extract the temporal and spatial features from two different types of sequences, we herein propose a model that consists of sub-networks based on RNN and CNN.

Our architectural model is comprised of two modules, i.e. CNN-RNN encoder and CNN decoder. The illustration of the proposed model is shown in Figure 6. In the encoder part, a combination of CNN and RNN is responsible for providing prior information during N time steps. It deals with two different input sequences. One is the sequential images of temperature distributions ($T_{t-N}, \dots, T_{t-1}$), while the other is the sequential values of operational conditions ($C_{t-N}, \dots, C_{t-1}$), which are obtained from the actual casting process, e.g., casting speed and the amount of cooling water. In the decoder part, a *deconvolutional* neural network, which is a CNN in the backward direction is employed for *upsampling*. Finally, the output of the network is a prediction value of the temperature distribution $\hat{T}_t$, one step ahead of the input sequence.

CNN in the encoder part produces a higher abstract representation of spatial information of the image data. As the input sequence, T in the form of stacked multiple arrays

are given to the network, it has hierarchically trainable layers of convolution operations, followed by a rectified linear unit (ReLU) as a nonlinear activation function at each layer. A layer of batch normalization is appended between the convolutional layer and ReLU, to perform the normalization for each training mini-batch [29]. The operation process with nonlinear activation is defined as:

$$T_r^n = \text{ReLU} \left(\sum_m T_m^{n-1} \otimes w_r^n + a_r^n \right) \quad (7)$$

where T_r^n is the output of r^{th} feature map in n^{th} convolutional layer, T_m^{n-1} is the m^{th} output of the previous layer. $\otimes$ represents convolution operation, whilst w_r^n and a_r^n denote weight and bias, respectively. After the feature maps are extracted via consecutive operation processes, an adaptive spatial pooling is finally implemented to enhance the feature representation of the image [30]. This is quite different from a conventional CNN, where fully connected layers and softmax classifiers are usually placed at the end of the network.

In the latter part of the encoder, the final feature representation from the CNN T_l is conditioned to RNN, redirecting the extracted one into RNN cell. Herein, T_l is additionally added for a hidden state and the RNN cell receives the input vector of sequential values of condition variables ($C_{t-N}, \dots, C_{t-1}$) every time step. Recurrence equation of the hidden state is expressed as:

$$h_k = \tanh(W_{C,h}C_k + W_{h,h}h_{k-1} + W_{E,h}T_l) \quad (8)$$

where h_k is the new hidden state of k^{th} time step, while C_k is the input value of k^{th} time step. h_{k-1} is the previous hidden state, which is generally initialized with zero at the beginning of the RNN. The $\tanh$ is used as a nonlinear activation function as in basic RNN cells. A weight matrix from CNN in the encoder to the hidden state, $W_{E,h}$ is added to describe how the information of the images is merged into the input sequence, in which the network learns the interactions between them while training the model.

A last hidden state that has a rich representation of image data T and condition values C is used as an input of CNN

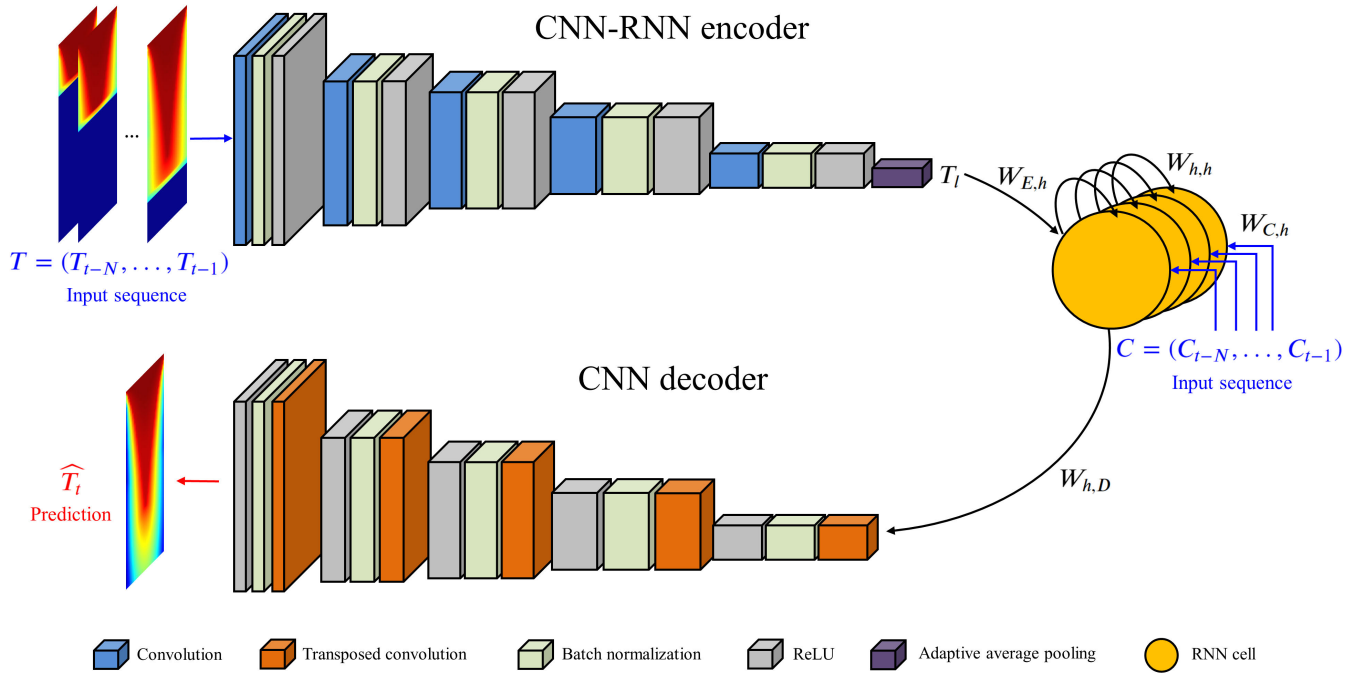


FIGURE 6. Proposed architectural model combining CNN-RNN encoder and CNN decoder.

TABLE 1. Proposed architecture of CNN in encoder and decoder module.

Layer	Type	Channels	Kernel size	Stride	Layer	Type	Channels	Kernel size	Stride
0	Input	3	-	-	6	2D Transposed Convolution	512	8×5	(1,1)
1	2D Convolution	32	3×3	(2,2)	7	2D Transposed Convolution	256	6×3	(2,1)
2	2D Convolution	64	3×3	(2,2)	8	2D Transposed Convolution	128	4×4	(2,2)
3	2D Convolution	128	3×3	(2,2)	9	2D Transposed Convolution	64	4×4	(2,2)
4	2D Convolution	256	3×3	(2,2)	10	2D Transposed Convolution	32	4×3	(2,2)
5	2D Convolution	512	3×3	(2,2)	11	Output	1	-	-

decoder, which generates the prediction image $\hat{T}_t$. Herein, transposed convolution layers are utilized in the replacement of convolution layers. Transposed convolution works with the opposite direction of the convolution operation to generate upsampled output by a sum over convolutions of the feature maps with trainable filters [31], [32]. As in the encoder part, there are consecutive transposed convolution layers with layers of batch normalization and nonlinear activation function that can be expressed as follows.

$$z_r^n = \text{ReLU} \left(\sum_m z_m^{n-1} \otimes f_r^n + b_r^n \right) \quad (9)$$

where z_r^n is the output of the process in r^{th} feature map, n^{th} is the transposed convolution layer, while the final value of z_r^n in the network becomes the prediction image. f_r^n , b_r^n are weight and bias learned during the training process, respectively.

Learning parameters depend on the proposed model are trained with a stochastic gradient descent algorithm for minimizing the error between the prediction value and the actual value. The loss function L of mean squared error in this

method is defined as:

$$L = \frac{\sum (T_t - \hat{T}_t)^2}{N} \quad (10)$$

where T_t and $\hat{T}_t$ are the actual and prediction image of the temperature distribution, respectively, while N is the number of training data set.

C. COMPARISON PROCEDURE

In this study, several procedures are deployed to validate the performance of the proposed model for predicting the temperature distribution in a continuous casting process. Several procedures such as comparison concerning types of recurrent cells, sequence length, and hyper-parameter settings are briefly outlined in this section.

First of all, the performances are compared among different types of recurrent cells, i.e., basic RNN, LSTM, and GRU. It is worth to be compared given that those recurrent cells show a significant difference in extracting the temporal information via hidden states, as introduced in Section II. Herein, not only the comparisons in the proposed model but the ones with baseline neural networks made up of those recurrent cells are conducted. Since the RNN-based standard

TABLE 2. Condition variables for analysis of temperature distribution.

Condition variables							
Material	Shape		Process				
Carbon content	Thickness of the steel	Width of the steel	Initial temperature	Ambient temperature	Casting speed	Mold heat flux	Cooling water segment 1~9

models are not designed to process the image, two-dimensional data of the temperature distribution are flattened to be fed into the models and trained for the benchmark. Considering that the flattening loses spatial information of the image, we investigate whether the proposed model using CNN for spatial features has an advantage or not. Besides, we are also interested to use bidirectional recurrent units [33] since in some cases, it helps us in making an accurate estimation about the dataset [34], [35].

Secondly, the influence of the length of input sequences, e.g. from time step $t - N$ to $t - 1$ on model performance is examined. It is shown that the performance of the RNN-based model decays as the sequence length increases, making it hard for the model to squeeze all sequential information of the input [36], [37]. Therefore, experiments are carried out to find the optimal sequence length N to obtain the best prediction accuracy. This is because there is a trade-off between the length and the capacity of feature representation. The prediction performances are compared concerning the three-level sequence lengths.

Finally, the performance of algorithms is assessed based upon the optimization of the hyper-parameters, including the number of hidden layers, hidden units, and dropout rates. The number of the hidden layers and hidden units is designed to find appropriate parameters of the network, while the dropout rate is determined to control *overfitting* as a way of regularization method. In particular, the dropout method for recurrent layers (denoted as RNN dropout) is used [38]. It is worth mentioning that such a method has successfully reduced the overfitting in several RNN-based approaches [39], [40].

D. PERFORMANCE MEASURES

The performance of the proposed method is evaluated with the performance of existing methods in terms of prediction accuracy and computation time. Prediction accuracy is measured using three different metrics, i.e. root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE). RMSE denotes a standard error between the predictions and actual values, whilst MAE shows an average of absolute differences between the actual values and the predictions. Lastly, MAPE can be considered as a weighted version of MAE, where for each instance, the absolute error is divided by the target value. The metrics are calculated as follows.

$$RMSE = \sqrt{\frac{\sum (T_t - \hat{T}_t)^2}{N}} \quad (11)$$

$$MAE = \frac{\sum |T_t - \hat{T}_t|}{N} \quad (12)$$

$$MAPE = \frac{1}{N} \sum \left| \frac{T_t - \hat{T}_t}{T_t} \right| \times 100\% \quad (13)$$

where T_t and $\hat{T}_t$ are the actual and the prediction value of the temperature distribution, respectively. N is the number of test data samples. Besides, the computation time in an actual casting process is compared to evaluate whether the proposed method is competitive on the dynamic control for predicting the temperature distribution in the process in comparison with the conventional mathematical method.

IV. EXPERIMENT RESULTS AND DISCUSSIONS

In this section, data description is firstly given, followed by the experimental results and discussions of the temperature prediction accuracy and computation time.

A. DATA DESCRIPTION

Operational condition values used to obtain the temperature distribution of solidifying steel are firstly described. Table 2 shows variables of operational conditions (denoted as condition variables), which can be classified into material, shape and process variables. The variables are casting speed, carbon content, ambient temperature and initial temperature (pouring temperature from tundish), as well as the shape of the steel, i.e., thickness and width. Furthermore, heat flux in the mold zone and the amount of cooling water at each segment in the spray zone are acquired as the key variable for primary and secondary cooling.

In this study, we obtained real-world data of the condition variables every second for analysis, which are produced by the steel-manufacturing company in Korea. There are in total 20 production cases of the continuous casting process and each case has a length of approximately 15,000 seconds which means there are around 15,000 sets of condition variables in a case. The entire data set is made up of 303,136 seconds, which are fully representative of the casting production cases. In the experiment, 16 cases (241,439 seconds) are used as a training set, while 4 cases (61,697 seconds) are used as a testing set. A collection of images representing the temperature distribution is extracted as a result of heat transfer and solidification analysis, as we described previously in Section II. Likewise, the same number of images are extracted with a dimension of 252×41 , representing the temperature distribution of the mold zone and the segments in the spray zone every second.

Finally, the data set is made by extracting collective subsequences from the entire data that are made up of the operational conditions C and the temperature distributions T in time series (see Figure 7). The input consists of several

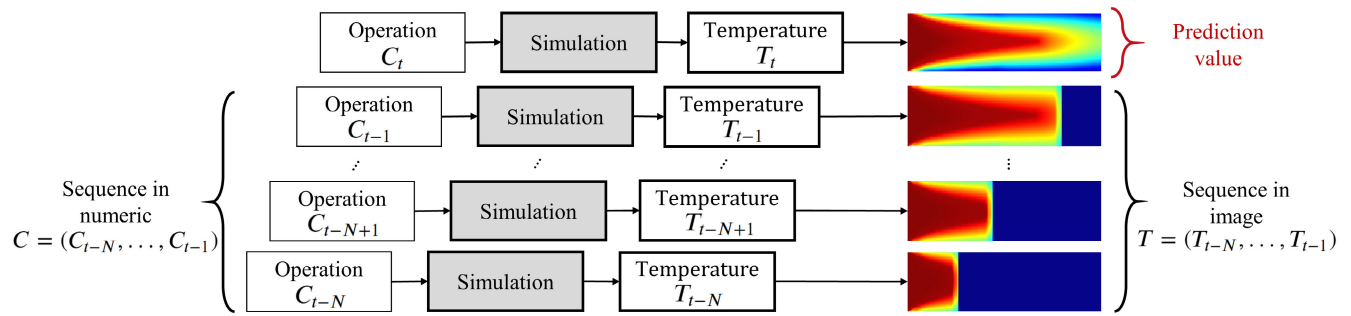


FIGURE 7. A method for sequential data generation.

TABLE 3. Performance comparison between baseline and proposed models (the best value is indicated in bold).

Prediction model	Recurrent cell	RMSE			MAE			MAPE (%)		
		3	6	9	3	6	9	3	6	9
Baseline models	RNN	43.93	48.75	58.77	30.89	33.12	43.31	10.40	10.65	12.38
	bi-RNN	40.79	44.65	49.20	29.49	31.31	33.93	9.31	9.56	10.69
	LSTM	41.15	45.01	51.90	30.19	32.42	35.99	8.95	9.09	9.80
	bi-LSTM	40.07	42.80	44.74	28.91	30.63	31.79	8.30	8.49	8.99
	GRU	42.52	43.18	50.20	30.31	31.89	34.18	9.12	9.20	9.79
Proposed models	bi-GRU	38.65	39.39	43.34	27.70	28.51	30.90	7.52	8.32	8.85
	RNN	14.68	17.60	18.59	11.32	11.78	11.83	4.24	4.38	4.54
	bi-RNN	12.79	14.51	14.04	8.80	8.93	9.35	3.38	3.39	3.53
	LSTM	11.90	13.65	13.79	8.55	8.87	9.04	2.98	3.34	3.49
	bi-LSTM	9.19	9.34	9.77	6.84	6.93	7.65	1.99	2.02	2.24
	GRU	12.55	15.41	15.27	8.74	9.05	9.21	3.01	3.35	3.52
	bi-GRU	9.81	13.67	12.81	7.82	8.11	8.28	2.08	2.23	2.31

*The results are presented with different sequence lengths of 3, 6 and 9 time steps

subsets of the operational conditions, as well as the subsets of the temperature distributions corresponding to the operational time step. The input from time steps $t - N$ to $t - 1$ becomes the sequences in numeric and image respectively, that have lengths of N . On the other hand, a prediction value T_t is a 252×41 matrix of the temperature distribution that is sequentially one step ahead of the sequence. Likewise, the whole data set has numerous pairs of the input sequences and the prediction values by sliding with a time window of a second.

B. PREDICTION OF TEMPERATURE DISTRIBUTION

Several algorithms are involved in the experiment to predict the temperature distribution. As depicted in Table 3, we categorize this comparative study into two main families, i.e., baseline models and proposed models. Each family possesses different kinds of models concerning different types of recurrent cells, i.e., RNN, bi-RNN, LSTM, bi-LSTM, GRU, and bi-GRU. Also, there is a parametric measure, i.e. sequence length, to specify the influence on the model's performance. Prediction models based on deep neural networks are trained to minimize the cost functions, while the hyper-parameters of each model are optimized via *grid search* method.

The quantitative results for temperature distribution prediction of the optimal models using baseline and the proposed

method are described in Table 3, which presents the mean values of the measurements for estimating the one-step-ahead temperature distribution on the testing set. Overall, it is obvious that the proposed method has a more accurate prediction performance than the baseline method. Depending on the type of recurrent cell, the models using LSTM and GRU in both families are likely to slightly outperform the ones with basic RNN cell. Furthermore, it is observed that the bidirectional unit of each model generally enhances the prediction accuracy, as the bidirectional unit performs in sequential dependencies of forwarding and backward direction.

In addition, it is shown that the length of the input sequences exerts influence on the prediction performance of each method. The longer sequence lowers the performance, preventing from capturing a piece of extended temporal information. However, the rise of the prediction performance with the bidirectional operation gradually increases as the sequence gets longer, owing to its characteristic of the two-way hidden state. Also, the models using RNN, LSTM, and GRU show relatively similar outcomes when the sequence is short, whereas the results from the models using LSTM and GRU become distinguishable from the ones using RNN cells.

In particular, more robust performance in terms of the sequence length appears in the proposed method.

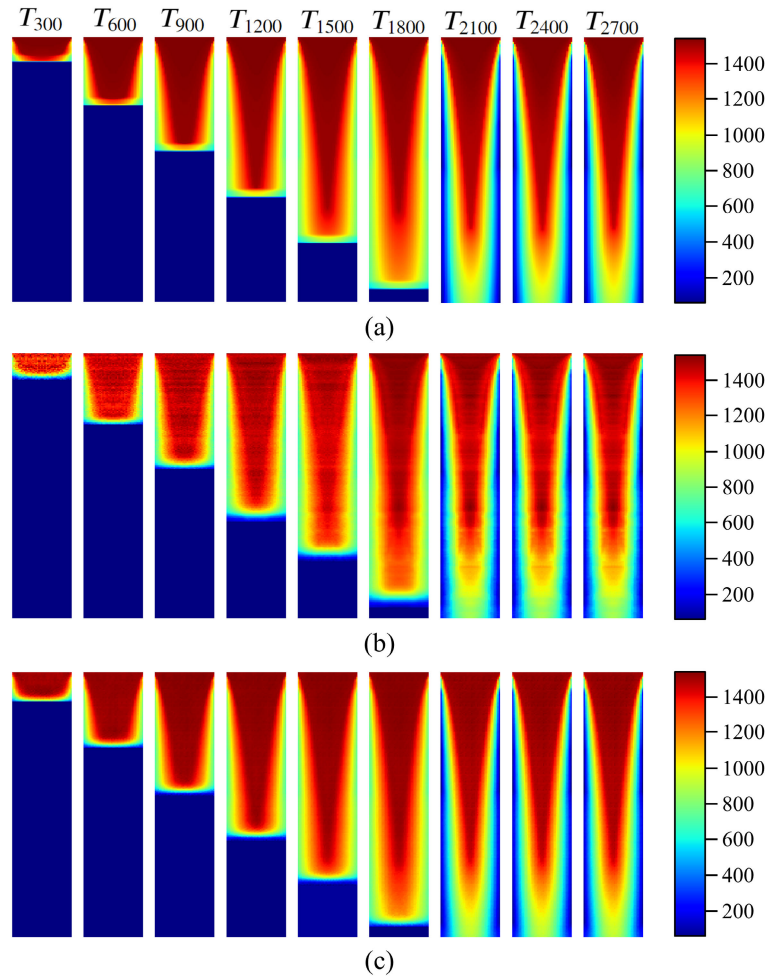


FIGURE 8. The temperature distribution of the solidifying steel in continuous casting: (a) actual values along the time step (b) predicted values of the baseline models using bi-GRU (c) predicted values of the proposed model using bi-LSTM.

The performance difference between the proposed models in terms of the sequence length is reasonably slighter than the baseline models. For example, the difference between the longest and the shortest sequence of the proposed model, e.g. bi-LSTM in terms of MAPE score is 0.25%, while the best one in the baseline models, e.g. bi-GRU is 1.33%. It can be said that the numerous input dimensions in the baseline models, i.e. temperature distribution and condition variables, make it burdensome for hidden states to seize the temporal dependency. On the other hand, the proposed method achieves more precise and robust performance via the encoder-decoder structure using the combination of CNN and RNN, not only by extracting spatial and sequential information but also by efficiently reducing the burden of capturing its information for deep neural networks.

To sum up, a promising approach for predicting the temperature distribution in a continuous casting process is obtained by the proposed method. The proposed model using bi-LSTM shows the lowest differences between the actual and estimated temperature by 6.84 degrees Celsius, while the

baseline model with bi-GRU reaches no less than 27.7 degrees Celsius, as shown in Table 3.

To further analyze the qualitative results for predicted images, the predictions of the temperature distribution from the testing data set are described in Figure 8. Each figure sequentially represents the actual temperature distributions and the predicted temperature distributions for the baseline and the proposed model, i.e., bi-GRU and bi-LSTM, respectively. Even though both models follow the actual trend, the image features of the predicted temperature distributions, e.g., a boundary line between solidus and liquidus, in the baseline model are not predicted precisely, meaning that spatial information of the temperature distribution is not successfully encoded. The prediction could not be properly utilized in an actual continuous casting process, due to its poor description on heat-transfer behavior of the solidifying steel, i.e., metallurgical length and solid fraction of the steel. In contrast, the proposed model that takes advantage of two different modules shows more plausible prediction results for both transient and steady-state phenomena of the steel.

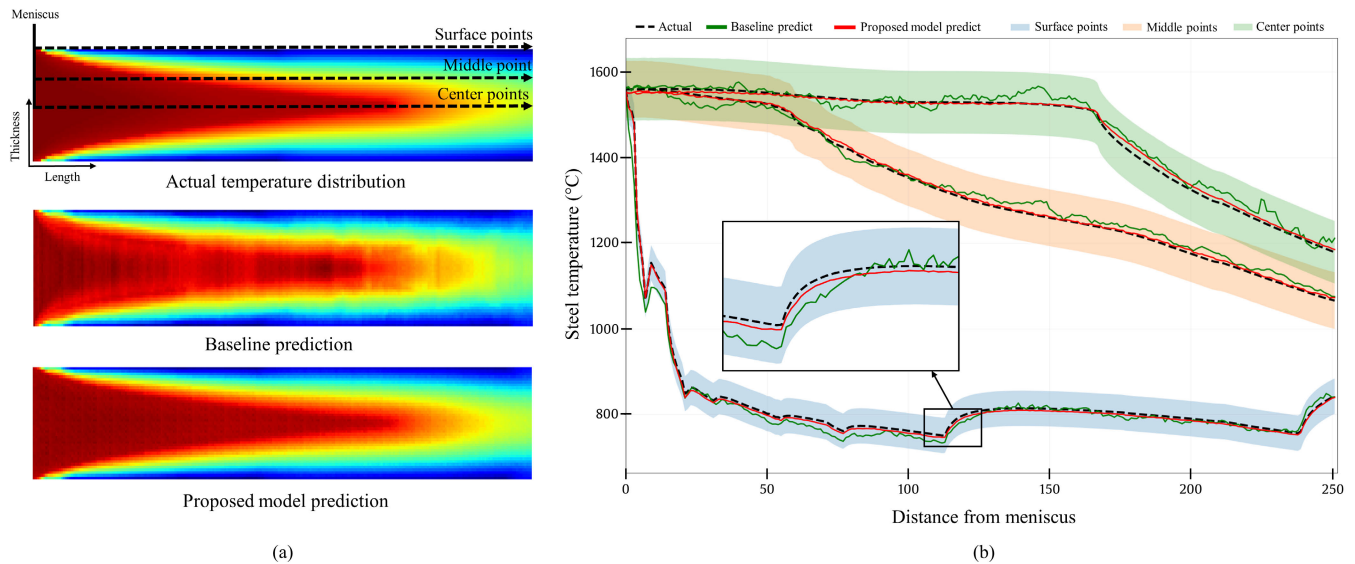


FIGURE 9. An example of temperature predictions of the solidifying steel: (a) actual and predicted temperature distributions at time step 2700, i.e. baseline model using bi-GRU and proposed model using bi-LSTM (b) actual and predicted temperature values along the distance from meniscus at the surface, middle and center point.

Furthermore, an example of temperature predictions at a certain time step among actual and estimated values is described in Figure 9. Figure 9 (a) shows true and predicted temperature profiles of the solidifying steel at time step 2700, while the values are forecasted by the optimal baseline and proposed model, i.e., bi-GRU and bi-LSTM, respectively. It is observed that the proposed model spatially yields the desirable prediction, compared to the actual heat-transfer image, while the baseline model brings about an improper prediction map without any consideration of spatial information. Given that the temperature distribution is a resulting temperature pattern obtained via primary and secondary cooling along with mold and spray zone, the unsuitable result of the baseline model would not be appropriate to be utilized for dynamic control of the continuous casting process. Figure 9(b) presents the actual and the predicted temperature data points at the surface, middle and center along the distance from meniscus since pixel values at each grid represent the temperatures within the steel along the length-thickness axis. The surrounding highlighted areas visualize marginal regions of a standard deviation from the actual temperature values. As expected, predicted values by the proposed model show a marked tendency to follow the trend of ground truth values in most of the temperature points, whereas the baseline model not only suffers from fluctuations but also worse estimation of the temperatures.

C. COMPUTATION TIME

We compare a deep neural network-based proposed model with a numerical analysis based model in terms of the computation time. We aim to develop not only a higher-performing prediction model but also a real-time model that could be used as an alternative for dynamic process control.

TABLE 4. Comparison of the computation time between the numerical analysis and proposed method.

Model	Time for a casting production case (seconds)	Time for one-step calculation (seconds)
Mathematical model	3994.82	0.45
Proposed model	3.46	0.0003

Table 4 shows the comparison result for each model in two different aspects, i.e., time for a casting production case and a 1-step calculation. The former indicates the time for an entire continuous casting process, where each case generally has over 15,000-time steps, while the latter is an average computation time consumed by a calculation of the temperature every time step. The results of the mathematical model are obtained by numerically calculating the temperature in the testing data set, whereas the ones of the proposed model constructed with deep neural networks show the inference time during the test.

It is shown that the computational cost of the proposed model is considerably lower than the mathematical model. Surprisingly, the mathematical model requires excessive computational resources by 3994.82 seconds to solve the temperature distribution during the whole process. On the contrary, the deep learning model needs less than 5 seconds to get the prediction. More promisingly, it is observed that only less than a millisecond is needed to predict a one-step-ahead temperature distribution via a deep neural network model. To summarize, our proposed method attributed to deep neural networks outperforms the conventional method concerning the computation time, which confirms the possibility of the proposed model for establishing a real-time dynamic control in an actual continuous casting process.

V. CONCLUSION

Temperature distribution of solidifying steel in a continuous casting process has a massive impact on product quality and operation safety, besides the process, i.e., primary and secondary cooling, should be properly controlled based upon the temperature distribution. Breaking away from the conventional model that is limited in an actual process, a deep neural networks based method for achieving precise and fast predictions of the temperature distribution was proposed in this paper. An integrative approach was presented to consider both historical temperature profiles and operational conditions, combining two modules of encoder and decoder that rely upon recurrent neural network (RNN) and convolutional neural network (CNN). The encoder was utilized to capture both spatial and sequential representations of the features, while trainable parameters of the decoder are optimized via *deconvolution* operations to predict one-step-ahead temperature distribution. The prediction performance of the proposed model was validated on real-world data obtained from the actual casting process, by comparing the performance of baseline models. It is observed that our proposed model greatly performed better than the baseline models, showing the best prediction error of 6.84 degrees Celsius. In particular, the temperature profiles estimated by the proposed model were nearly identical to the actual values, induced by spatial feature extraction of the images. Furthermore, our suggested model showed increased feasibility of the model being utilized in the actual continuous casting for dynamic process control, as it earned an extremely fast computation time for predicting the profile at each step and each production case, compared to the conventional numerical model.

REFERENCES

- [1] H. Wang, G. Li, Y. Lei, Y. Zhao, Q. Dai, and J. Wang, "Mathematical heat transfer model research for the improvement of continuous casting slab temperature," *ISIJ Int.*, vol. 45, no. 9, pp. 1291–1296, 2005.
- [2] J. Ha, J. Cho, B. Lee, and M. Ha, "Numerical analysis of secondary cooling and bulging in the continuous casting of slabs," *J. Mater. Process. Technol.*, vol. 113, nos. 1–3, pp. 257–261, Jun. 2001.
- [3] S. Louhenkilpi, "Continuous casting of steel," in *Treatise Process Metallurgy*. Amsterdam, The Netherlands: Elsevier, 2014, pp. 373–434.
- [4] A. Ray, S. Ray, D. Basu, and S. Mazumdar, "Continuous casting: A review," *Steel India*, vol. 26, no. 3, pp. 138–154, 2004.
- [5] J. Sengupta, B. Thomas, and M. Wells, "The use of water cooling during the continuous casting of steel and aluminum alloys," *Metall. Mater. Trans. A*, vol. 36, no. 1, pp. 187–204, 2005.
- [6] S. Chaudhuri, R. K. Singh, K. Patwari, S. Majumdar, A. K. Ray, A. K. P. Singh, and N. Neogi, "Design and implementation of an automated secondary cooling system for the continuous casting of billets," *ISA Trans.*, vol. 49, no. 1, pp. 121–129, 2010.
- [7] J. Brimacombe, "Design of continuous casting machines based on a heat-flow analysis: State-of-the-art review," *Can. Metallurgical Quart.*, vol. 15, no. 2, pp. 163–175, 1976.
- [8] S. Mazumdar and S. K. Ray, "Solidification control in continuous casting of steel," *Sadhana*, vol. 26, nos. 1–2, pp. 179–198, Feb. 2001.
- [9] J. Zhang, D. Chen, S. Wang, and M. Long, "Compensation control model of superheat and cooling water temperature for secondary cooling of continuous casting," *Steel Res. Int.*, vol. 82, no. 3, pp. 213–221, Mar. 2011.
- [10] M. J. Long, D. F. Chen, Q. X. Wang, D. H. Luo, Z. W. Han, Q. Liu, and W. X. Gao, "Determination of CC slab solidification using nail shooting technique," *Ironmaking Steelmaking*, vol. 39, no. 5, pp. 370–377, Jul. 2012.
- [11] T. Kawawa, H. Sato, S. Miyahara, T. Koyano, and H. Nemoto, "Determination of solidifying shell thickness of continuously cast slab by rivet pin shooting," *Tetsu-Hagané*, vol. 60, no. 2, pp. 206–216, 1974.
- [12] Y. Meng and B. G. Thomas, "Heat-transfer and solidification model of continuous slab casting: Con1d," *Metall. Mater. Trans. B*, vol. 34, no. 5, pp. 685–705, 2003.
- [13] R. Davies, N. Blake, and P. Campbell, "Solidification modelling—An aid to continuous casting," in *Proc. 4th Int. Conf. Continuous Casting. Preprints.*, vol. 2, 1988, pp. 645–654.
- [14] B. G. Thomas and L. Zhang, "Mathematical modeling of fluid flow in continuous casting," *ISIJ Int.*, vol. 41, no. 10, pp. 1181–1193, 2001.
- [15] R. Pehlke, A. Jeyarajan, and H. Wada, "Summary of thermal properties for casting alloys and mold materials," Univ. Michigan, Ann Arbor, MI, USA, NTIS, Tech. Rep. NSF/MEA-82028, 1982.
- [16] G. E. Totten, K. Funatani, and L. Xie, *Handbook of Metallurgical Process Design*. Boca Raton, FL, USA: CRC Press, 2004.
- [17] J. Savage and W. Pritchard, "The problem of rupture of the billet in the continuous casting of steel," *J. Iron Steel Inst.*, vol. 178, no. 3, pp. 269–277, 1954.
- [18] T. Nozaki, J.-I. Matsuno, K. Murata, H. Ooi, and M. Kodama, "A secondary cooling pattern for preventing surface cracks of continuous casting slab," *ISIJ Int.*, vol. 18, no. 6, pp. 330–338, Jun. 1978.
- [19] Z. Sterjovski, D. Nolan, K. Carpenter, D. Dunne, and J. Norrish, "Artificial neural networks for modelling the mechanical properties of steels in various applications," *J. Mater. Process. Technol.*, vol. 170, no. 3, pp. 536–544, Dec. 2005.
- [20] C. A. Santos, E. L. Fortaleza, C. R. F. Ferreira, J. A. Spim, and A. Garcia, "A solidification heat transfer model and a neural network based algorithm applied to the continuous casting of steel billets and blooms," *Model. Simul. Mater. Sci. Eng.*, vol. 13, no. 7, pp. 1071–1087, Oct. 2005.
- [21] W. Choi, H. Huh, B. A. Tama, G. Park, and S. Lee, "A neural network model for material degradation detection and diagnosis using microscopic images," *IEEE Access*, vol. 7, pp. 92151–92160, 2019.
- [22] G. W. Song, B. A. Tama, J. Park, J. Y. Hwang, J. Bang, S. J. Park, and S. Lee, "Temperature control optimization in a steel-making continuous casting process using a multimodal deep learning approach," *Steel Res. Int.*, vol. 90, no. 12, Dec. 2019, Art. no. 1900321.
- [23] A. M. Rather, A. Agarwal, and V. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 3234–3241, Apr. 2015.
- [24] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Academic Annu. Conf. Chin. Assoc. Automat. (YAC)*, Nov. 2016, pp. 324–328.
- [25] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018.
- [26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [27] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: <https://arxiv.org/abs/1406.1078>
- [28] Y. Lecun and Y. Bengio, "Convolutional networks for images, speech, and time series," *Handbook Brain Theory Neural Netw.*, vol. 3361, no. 10, p. 1995, 1995.
- [29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 448–456.
- [30] Y. Liu, Y.-M. Zhang, X.-Y. Zhang, and C.-L. Liu, "Adaptive spatial pooling for image classification," *Pattern Recognit.*, vol. 55, pp. 58–67, Jul. 2016.
- [31] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2016, *arXiv:1603.07285*. [Online]. Available: <https://arxiv.org/abs/1603.07285>
- [32] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2528–2535.
- [33] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
- [34] J. C.-W. Lin, Y. Shao, Y. Zhou, M. Pirouz, and H.-C. Chen, "A Bi-LSTM mention hypergraph model with encoding schema for mention extraction," *Eng. Appl. Artif. Intell.*, vol. 85, pp. 175–181, Oct. 2019.
- [35] B. Cortez, B. Carrera, Y.-J. Kim, and J.-Y. Jung, "An architecture for emergency event prediction using LSTM recurrent neural networks," *Expert Syst. Appl.*, vol. 97, pp. 315–324, May 2018.

- [36] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: <https://arxiv.org/abs/1409.0473>
- [37] G. C. Sergio, D. S. Moirangthem, and M. Lee, "Temporal hierarchies in sequence to sequence for sentence correction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–7.
- [38] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1019–1027.
- [39] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, Oct. 2018.
- [40] A. T. Müller, J. A. Hiss, and G. Schneider, "Recurrent neural network model for constructive peptide design," *J. Chem. Inf. Model.*, vol. 58, no. 2, pp. 472–479, Feb. 2018.



SOO YOUNG LEE received the B.S. degree from Chung-Ang University, Seoul, South Korea, in 2019. He is currently pursuing the M.S. and Ph.D. degrees with the Industrial Artificial Intelligence Laboratory, POSTECH, South Korea. His research interests include industrial artificial intelligence with mechanical systems and applications of the AI-based smart manufacturing.



BAYU ADHI TAMA received the Ph.D. degree in IT convergence engineering from Pukyong National University, South Korea. He was a Postdoctoral Researcher with the School of Management Engineering, Ulsan National Institute of Science and Technology, South Korea. He is currently a Postdoctoral Researcher with the Industrial Artificial Intelligence Laboratory, POSTECH, South Korea. He has published more than 30 articles in academic journals and conferences. His research interests include machine learning and artificial intelligence techniques applied for manufacturing, cybersecurity, and medical informatics. He received a Full Scholarship for his Ph.D. degree from the Korean Government and the Award for his Excellent Academic Achievement from the Ministry of Education, South Korea.



CHANGYUN CHOI received the B.S. degree from Soongsil University, Seoul, South Korea, in 2019. He is currently pursuing the M.S. degree with POSTECH, South Korea. His research interests include application of artificial intelligence in robotics.



JONG-YEON HWANG received the B.S., M.S., and Ph.D. degrees from the Department of Mechanical Engineering, Inha University, Incheon, South Korea. He was a Postdoctoral Research Fellow with the University of Illinois Urbana-Champaign, Champaign, IL, USA. He is currently a Principal Researcher with the POSCO Process and Engineering Research Laboratory, Gwangyang-si, South Korea, where he is also in charge of the development of continuous caster and its cooling systems. His research interest includes the design of a continuous casting system for the next generation.



JONGGEUN BANG received the B.S. and Ph.D. degrees from the Department of Mechanical Engineering, Yonsei University, Seoul, South Korea. He is currently a Senior Researcher with the POSCO Process and Engineering Research Laboratory, Gwangyang-si, South Korea, where he is also in charge of designing cooling system of continuous casting with the Thermo-Fluid and Process Research Group. His research interest includes design and optimization of continuous casting cooling systems.



SEUNGCHUL LEE received the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 2008 and 2010, respectively. He was an Assistant Professor with the Ulsan National Institute of Science and Technology, South Korea, and a Postdoctoral Research Fellow with the University of Michigan. He has been an Assistant Professor with the Department of Mechanical Engineering, POSTECH, since 2018. His research interests include industrial artificial intelligence with mechanical systems, deep learning for machine healthcare, the IoT-based smart manufacturing, and manufacturing data visualization. He extends his research work to developing self-sustainable systems via an intelligent, informatics, and the industrial IoT system design at POSTECH.

...