CHAPTER

16

Predicting temperatures inside a steel slab reheating furnace using Deep Learning

https://doi.org/10.56238/alookdevelopv1-016

Rodrigo de Souza Lima
Master's student in Control and Automation Engineering, Federal Institute of Espírito Santo, Serra, EspiritoSanto
E-mail: Rodrigo.silima@arcelormittal.com.br

Leonardo Azevedo Scárdua
PhD in Electrical Engineering, Federal Institute of Espirito Santo Serra, EspiritoSanto
E-mail: lscardua@ifes.edu.br

Gustavo Maia de Almeida
PhD in Electrical Engineering, Federal Institute of Espirito Santo, Serra, EspiritoSanto
E-mail: gmaia@ifes.edu.br

ABSTRACT

Due to the complexity and high financial costs involved in production processes, the steel industry can benefit from applications of intelligent systems, capable of performing automated activities. This research paper addresses a description of the process of creating a data-driven computational system to develop a computational thermal model of a real steel plate reheating furnace. Sufficiently accurate computational models can be used in conjunction with combustion control optimization techniques, such as model-based predictive control (MPC), or even a Digital Twin of the combustion system of a plate reheating furnace. The tool can be used in predictive failure diagnosis, fundamental for the maintenance and operation teams responsible for asset management. For this development, Recurrent Artificial Neural Networks have been widely applied, validating the existence of series that have temporal links between their samples, a typical case of monitoring industrial process variables. To meet the proposed objective, the performance of models based on recurrent neural networks of the Long Short Term-Memory (LSTM), Gated Recurrent Unit (GRU), and Temporal Convolutional Network (TCN) type was analyzed. The results were evaluated under different prediction horizons, since such techniques demand models capable of accurate predictions that are several steps ahead, premised on prediction capability.

Keywords: Furnace, Model neural l, Prediction ahead, Temperature, Reheating, Steel slab.

1 INTRODUCTION

The reheating furnaces aim to heat steel plates in a controlled way, at temperatures close to 1200°C, for their proper processing in the hot rolling of strips [1]. Modern reheating furnaces are dynamic, nonlinear, and complex systems with heat transfer performances that can be greatly influenced by operating conditions [2]. Heating must be homogeneous to obtain the highest productivity, lowest possible energy consumption, and lowest rates of pollutant emissions [3].

The temperature of the plates cannot be measured directly inside the furnace and is inferred by mathematical models of heating, which are not trivial, due to factors such as the complexity of the parameters, thermal hysteresis, and the strong coupling existing between the heating zones [4]. The process of reheating steel plates is responsible for approximately 15 to 20% of the total energy consumed in a steel mill and for 70% of the energy consumed in the rolling process [5].

In this context, there is a bias for the application of control systems that guarantee the maximum performance of the equipment. Intelligent systems based on computational models can be applied in the reliable reproduction of the thermal dynamics of a reheating furnace. The tool can be useful for explaining heating optimization problems or even predicting combustion system failures. Applications of deep learning neural networks have been studied to solve the problem of energy consumption in furnaces. [6] established a prediction model based on Gated Recurrent Unit (GRU) type recurrent networks to predict the internal temperature of the furnace using time series d the temperatures, fuel, and air. An interesting proposal of temperature prediction in a reheating furnace, using a convolutional temporal network with learning transfer between the zones of a furnace heating [7]. Already [8], proposed a new method of predicting furnace temperature using the optimized kernel extreme learning machine (OKELM). [9] It presents a new approach to the multivariate linear regression spatiotemporal variable parameter zoning model (MLR-VPST) applied to temperature prediction.

Knowing that the s temperature s inside a plate reheating furnace can m be predicted s due to the temporal characteristic, this research proposes the neural modeling of seven zones of heating of mobile beam-type furnaces and pulsating burners using deep learning networks. The latter is a differential of the research, since in this application there are no continuous measurements of the air and gas flows per burner, increasing the challenge in the process of thermal modeling of the heating zones. The goal is to have accurate models capable of making predictions step ahead.

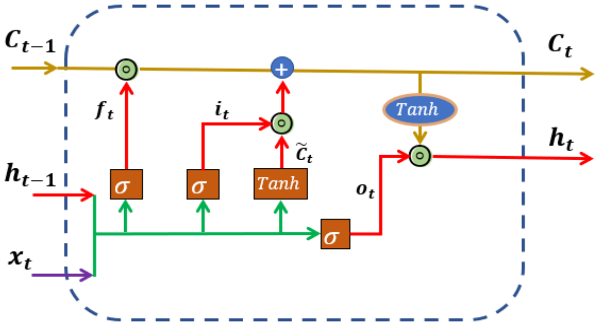
1.1 DEEP LEARNING NEURAL NETWORKS

1.1.1 Long-Short Term Memory

This type of neural network was introduced by [10] for the explicit purpose of helping to solve the unstable gradient problem [1 1]. The long-term memory network, usually called LSTM, is a special type of recurrent network, capable of learning long-term connections. This recurring network model can store larger amounts of information, pertaining to past inputs, while maintaining the relevance of recent states.

In short, the architecture of LSTMs consists of a set of memory cells connected in a recurring way. A cellular unit, see "Fig. 1", is composed of four gates, which in turn are activated by mathematical functions.

Fig. 1. LSTM Cell



The central point of LSTMs is the state of the cell. The main horizontal line (C_t) that passes through the top of the diagram is known as long-term memory and is responsible for carrying the state of the cell, traversing the entire chain with only a few linear interactions. The inputs of the cells are considered augmented, due to the concatenation of the new information and x_t h_(t-1) that represents the output of the previous state. The interactions of sigmoid functions and element-by-element vector multiplication operations function as gates, adding or removing information to the cell state. In the initial part of the cell, we have the Forgot Gate seen in (1), where the output is a sigmoid function that controls what information will be forgotten.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f \qquad (1)$$

In the central part, the Input Gate unit appears or the input section (2), but in this particular case, a hyperbolic tangent function (3) calculates a possible context, and the Cell Up unit (4) controls the addition of this new context to the memory unit.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i \qquad (2)$$

$$\widetilde{C}_t = tanh(W_C.[h_{t-1}, x_t] + b_C \qquad (3)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \qquad (4)$$

Finally, the Output Gate (5), controls the information that follows for outpu 〖_t〗 , being also, the input of the next cell in time 〖_(-1)〗 .

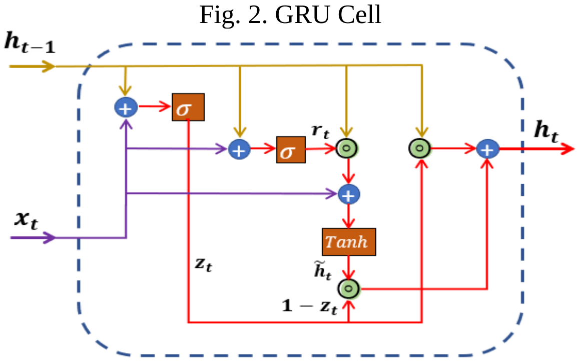
$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o \qquad (5)$$

1.2 GATED UNIT RECURRENT

The GRU is a type of recurrent neural network that uses so-called update and restart gates. They are two vectors that decide what information should be passed to the output. What's special about them is that they can be trained to store information from a long time ago, without washing it away in time or removing information irrelevant to the forecast [12].

As proposed by [13] the GRU makes each recurring unit adaptively capture the dependencies of different time scales. The flow of information is modulated by the "gates" in a similar way to LSTMs (Long-Short Term Memory), within the cellular unit, but without having separate memory cells. "Fig. 2" features a GRU cell unit.

Fig. 2. GRU Cell



The mathematical operations of the inner workings of the GRU recurring unit are presented below. The Update Gate memory updates control gate is calculated by (6). Where, is the x_t input of new information, and is the output of the previous state that contains the past information? Both are multiplied by their respective weights 〖_(-1)〗 sW_t and. The result is summed and then activated by a sigmoid function 〖 U〗 _t.

$$z_t = \sigma(W_t x_t + U_t h_{t-1}) \qquad (6)$$

The Update Gate supports the cellular unit that determines the amount of past information that will be retained, eliminating the risk of the gradient disappearing [14]. The gate of oblivion, Reset Gate, is controlled by (7).

$$r_t = \sigma(W_t x_t + U_t h_{t-1}) \qquad (7)$$

The information calculated and contained in h ̃_t (8), are possible candidates to enter the state of memory. Baseada in the new entries and x_t n the gate of oblivion, which in turn will allow or not the information of the previous state to be added in the present state 〖_〗 .

$$\tilde{h}_t = tanh(W x_t + U(r_t \odot h_{t-1})) \qquad (8)$$

Finally, the current memory h_t will be updated in Update Memory (9).

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t \qquad (9)$$

1.3 TEMPORAL CONVOLUTIONAL NETWORK

The convolutional empirical TCN (Temporal Convolutional Network) is quite efficient in applying sequential tasks to the test. This type of deep learning network employs circumvolutions and casual dilations in its structure, so it is suitable for sequential data with temporality [15].

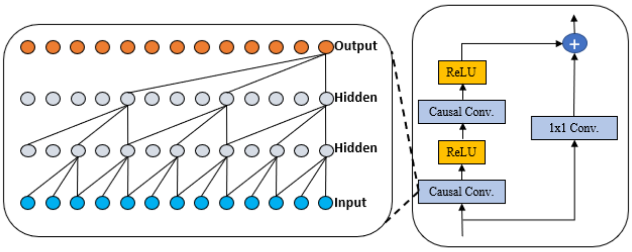
Some characteristics define TCN. Calculations are performed without layers, which means that each time step is updated simultaneously, rather than updating sequentially by frame. The architecture can take a sequence of any length and map it to an output sequence of the same length. Convolutions are causal, this means that there is no "leakage" of information from the future to the past. The operation of a TCN is based on two principles: the network uses a fully convolutional one-dimensional architecture, where each hidden layer is of the same length as the input layer [16]. A mask for fills with zeros is applied to each hidden layer to keep subsequent layers at the same length as the previous ones [17]. In summary, TCN is the sum of FCN 1D + causal convolutions.

The expertise of TCN is in the application of dilated convolution, which is a convolution where a filter is applied over an area larger than its length, skipping input values with a certain step, ensuring that the network covers more information with the increase in the size of the receptive field. According to the works of [1 8] [1 9] the dilated convolutions result in the exponential growth of the receptive field with only a few layers, that is, the number of delays that the network can see to predict its output, preserving the input resolution in every network, as well as computational efficiency. Formally, for a one-dimensional sequence of input } {x and a filter the operator { ∈ R^n f:{0,…,k-1}→R, of the dilated convolution applied to sequence {s} can be definedF} as (10):

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i).x_{s-d.i} \qquad (10)$$

Where, is the d dilation factor, is the size of the filter,k ands-d.I explain the direction that the convolution will be applied. If it is equal to 1 the dilated convolution is reduced to a regular convolution. The receptive field in the TCN is directly linked to the depth of the network, as well as the size of the filter and the dilation factor. To prevent the deeper network structure from complicating the learning process, a residual connection "Fig. 3" is added to the output. Because the input and output can have different widths, the connected residue uses a 1 x 1 convolution to ensure that the addition operation receives the same tensors [20].

Fig. 3. Dilated causal convolution employed in the residual block

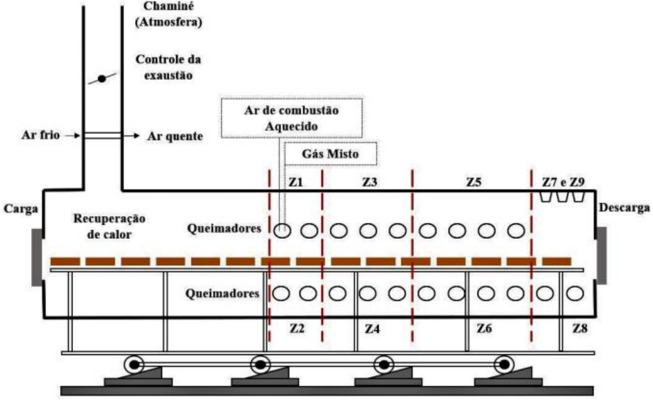


The Rectified Linear Unit (ReLU), is an activation function that returns to 0 if it receives any negative input, but for any positive value { it returns x}to the same value as the input.

2 STUDY EQUIPMENT

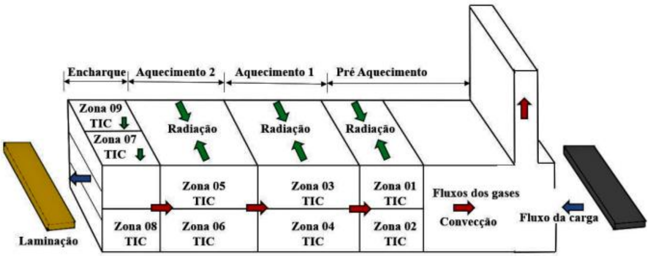
With a production capacity of 400 tons per hour and divided into nine heating zones, control regions of has temperature without a physical barrier, with a total of 20 pairs of side burners and 21 ceilings, the furnace studied here, uses as fuel industrial gas, which is a mixture of steel gases, with controlled calorific value. As an oxidizing agent, atmospheric air is used, which is reheated in a heat recovery system. The transport of the plates inside the furnace is using walking beams (Walking Beam). "Fig. 4" presents a side view of the furnace and its main components. The plates move from loading to dewatering in a controlled manner as specified for thermal purposes. The flow of the gas ex post-combustion is contrary to the flow of the plates and controlled by an existing valve in the exhaust channel. A heat recovery system is installed in the exhaust of the gases, to heat the combustion air.

Fig. 4. Side view of the oven and its components



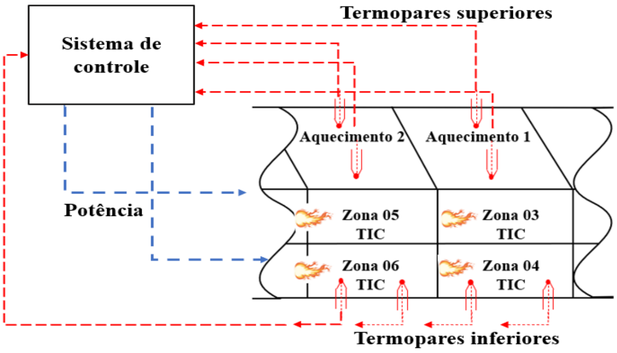
The furnace has three heating zones (Z), subdivided into upper and lower "Fig. 5". Heat is transferred to charge using conduction and radiation, where the mechanism of radiation is predominant [21]. Each zone has its dedicated temperature controller (TIC) and the temperature rise occurs gradually along the furnace, where each heating zone has its thermal objective.

Fig. 5. Zones heating and system dynamics.



In each heating zone, temperature variations are measured by thermocouple-type sensors and sent to the thermal model, which estimates the temperature of each product inside the furnace. Thus, the optimized control model is fed back with the thermal energy application demand, according to "Fig. 6". The burners are individually actuated using an On/Off control valve. The setup for each burner is sent on a time basis, and a frequency-modulated technique (pulse flame) is applied [2 2], so that the burners are switched between the two states, off or on, based on a cycle of determined work, which controls the input of heat to the process. The input heat demand determines the imposed cycle, which is the time when the burner will be on or off.

Fig. 6. Control system configuration



3 SYSTEM MODELING

To construct the forecast model, data were collected from a plate reheating furnace after maintenance. Where it was carried out The cleaning of the gas pipes, orifice plates, valves, and calibration of the burners. All on-off control zones were modeled. The data sets were individually processed for each control zone of temperature.

3.1 BUILDING THE DATABASE

The system information for building the database for training and testing was collected between 08/13/2022 and 08/14/2022 through the IbaAnalyzer software. The sampling time was 1 second, fast enough to capture the dynamics of the furnace air, gas, and exhaust flows. Each dataset has 43,201 samples. Before any manipulation of the data, an exploratory analysis was previously performed to evaluate the consistency of the data. Avoiding, working by contaminated or missing data.

The selection of the input variables was based on the operational behavior of the furnace, on the tacit knowledge of the process, and through statistical analysis of autocorrelation, which made it possible to reduce 23% of the database of predicted characteristics, due to the low correlation with the variable of interest or strong correlation with each other. The technique was useful for database optimization.

Some variables are common to all zones of the furnace, but each zone has unique input variables. Common variables are general airflow, general fuel flow, furnace exhaust flow and valve position, and exhaust control of the gases to the chimney. The individual variables for each zone are the s powers applied to each burner zone and the temperature of the adjacent zone, following the flow of gases. The total value of the input variables per model was modified according to the number of burners. The data were separated into 70% for model training and 30% for validation.

3.2 NORMALIZATION OF DATA

To achieve high performance in machine learning, data must be pre-processed for normalization [2, 3]. Here, all data were subjected to pre-processing, since the values of the chosen variables are in different ranges. There are several well-known normalization techniques like Simple Feature Scaling, Min-Max, Z-score, etc. The preprocessing strategy was the same for the input and output variables of the combustion system and consisted of standard normalization, which centralizes the data on the zero mean with a standard deviation of 1z (11). Where, are the data to be observed, μ is the mean and σ the standard deviation. X

$$z = \frac{X - \mu}{\sigma} \qquad (11)$$

3.3 DATA PREPARATION FOR KINGDOM SUPERVISED

Through historical data, models based on deep learning neural networks can develop a functional relationship between the input resources and the future values of the objective variable. The resulting model can provide predictions about the objective variable at time points in the future. The learning of the models is based on the functional relationship between the sis temporal rie and can be expressed mathematically by (12). Where is the 〖_〗 input characteristic vector observed in time and is the prediction yˆ_(t+k)at a future time point?

$$\hat{y}_{t+k} = f_k\,(x_{k-w}, \dots, x_{t-1,}\, y_{k-w}, \dots, y_{t-1}) \qquad (12)$$

Wherey ̂_(t+k) and prediction of the objective variable for the time t+k. k is the period in the future for which the target variable is to be predicted. They are already $y_{k-w}, \dots, y_{t-1}$ the target values observed from time $t-w$ to $t-$,. $x_{k-w}, \dots, x_{t-1}$ are the characteristic vectors of inputs, observed from time $t-w$ a $t-1$. f_k It is the function trained by deep learning models.

For the supervised training of temporal prediction models, it is necessary to prepare the database containing the historical series in advance. The entry in the model of have a uniform length [24]. A fixed-size sliding time window applies. Predictions of future data can be one step forward or several steps forward. The sequence is temporal (13) and exemplifies a fixed look-back window in 4 past times with a prediction of n steps ahead.

$$x(t-3), x(t-2), x(t-1), x(t) \rightarrow \hat{y}\,(t+n) \qquad (12)$$

3.4 ARCHITECTURES

In this scientific research, three types of deep learning neural networks were used, two recurrent LSTM and GRU, and one onvolutional C (TCN). The architecture and configuration of the networks were maintained similarly, in the number of layers, training optimizer, and learning rate. The objective was to evaluate the performance between the proposed networks as a temperature prediction tool on the same basis of comparison.

The tests were performed with the length of the "look-back" sliding window 25 steps back. The predictions were made with a single step, 60 and 120 steps ahead. Each step represents 1 second. The prediction horizons were chosen based on the operating time of the burners. For the loss function we use the EAM and as an optimizer the Adam algorithm, which is an optimization method based on the descent of the stochastic gradient based on the adaptive estimation of first and second-order moments [25].

3.5 PERFORMANCE METRICS

To measure the performance indices, normally, the errors that it presents are analyzed, that is, the comparison of the actual output with the value predicted by the model in the same instant of time, in the case of time series. Both the mean square root of error (RMSE) and the absolute mean error

(EAC) are regularly employed as performance evaluation criteria [27]. Sendo these metrics used here as evaluative parameters, where: best value = 0; worst value = +∞.

The absolute mean error, MAE is calculated from the mean of the absolute errors, that is, we use the module of each error to avoid underestimation, because the value is less affected by especially extreme points (outliers). Each error can be interpreted as the difference between the actual value and the predicted value (15).

$$MAE = \frac{1}{n_{ts}} \sum_{i=0}^{n_{ts}} |yi^{\{ts\}} - \hat{y}_i| \qquad (15)$$

The square root of the mean error, RMSE (16) is often used in time series because it is more sensitive to larger errors, due to the quadrature process that produced it.

$$RMSE = \sqrt{\frac{1}{n_{ts}} \sum_{i=0}^{n_{ts}} (\hat{y}_{i-} yi^{\{ts\}})^2} \qquad (16)$$

4 FINDINGS

The results presented now refer to the seven "Z" warming zones, with predictions of 1, 60, and 120 steps ahead. The results are presented in Table I to provide a quick visualization of the performance of each model. The amount of graphic information generated was limited, so it was decided to graphically present the results of zone 3 because the behavior of the rest was similar.

Table 1: Results of Predictions

Z	**Model**	**Step 1**		**60 steps**		**120 steps**	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
1	GRU	1,262	1.710	2,730	3.7 56	4.1 63	5,782
	LSTM	2,744	3,385	4.999	5,898	7,385	9.172
	TCN	4,394	6,106	4.4 32	6,706	5.2 82	8,536
2	GRU	1,061	2.319	1,757	2,372	2,108	2,936
	LSTM	2,971	3,607	4.0 2 2	4,719	5,488	6.267
	TCN	3,774	5,155	3,298	4,821	3,551	5,223
3	GRU	1,343	1,861	3,031	3.997	4,689	5,986
	LSTM	2,833	3,515	5,392	6,183	7,908	9,139
	TCN	3,162	4,048	3,621	4,784	4,697	6.346

4	GRU	0.808	1,064	1.059	1,360	1.682	2,138
	LSTM	1,718	2,212	1.998	2,522	2,601	3,134
	TCN	2,482	3,183	3.007	3,642	3,950	4,688
5	GRU	1.083	1,410	1,062	1,404	2.186	2,633
	LSTM	3.232	4,435	3,623	5.287	4,309	6,045
	TCN	3.0 33	3,976	3,657	4.3 24	4.386	5.3 65
6	GRU	0.898	1.101	0.99	1,177	1,495	1,870
	LSTM	3,012	3,805	3,688	4,506	4.356	5.224
	TCN	3,102	4,196	3,623	4,818	4,611	5.8 58
8	GRU	0.331	0.4 2 2	0.408	0.462	0.415	0.552
	LSTM	1,690	2,104	1,884	2.280	2,062	2.490
	TCN	1,141	1,411	1,260	1.5 2 2	1,539	1.8 63

This 'fig.' (8), (9), and (10), present the results of zone 3 at a predetermined period within the test data. Repetitively the figures are arranged in 1 step forward, 60 steps, and 120.

Fig. 8. Zone 3 temperature with 1 step forward

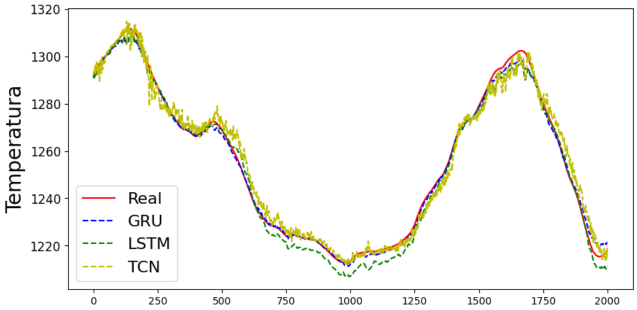


Fig. 9. Zone 3 temperature with 60 steps ahead

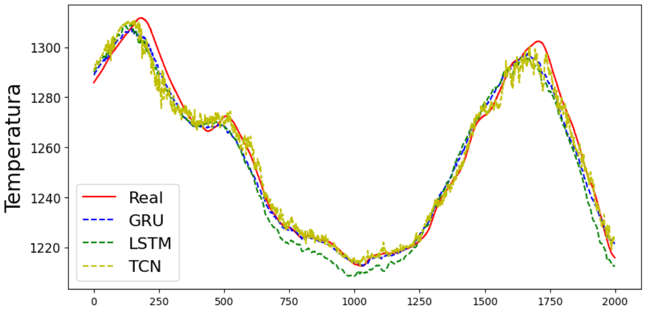
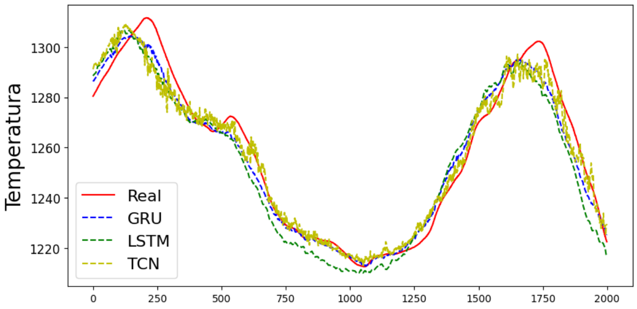


Fig. 10. Zone 3 temperature with 120 steps ahead



In general, we see a positive performance of the models for this application, since in the z worst cases, the errors in the predictions did not exceed 10°C, with emphasis on the GRU network that reached the lowest MAE and RMSE values in all horizons. It is observed that the farther away the prediction horizon is, the more degradation of the models increases. Some points of emphasis are observed in the prediction of the model of zone 08, where the results are satisfactorily superior to those of the other zones. In this case, even at such distant horizons, the predictions were accurate. There is a sharp variation in the predictions of TCN, but it is believed to be possible to adjust through adjustments of the network control parameters.

5 CONCLUSION

This paper described the creation of computational models based on deep learning neural networks, applied in predicting the internal temperature of a real steel plate reheating furnace that uses pulsating burners in its thermal control system. The approach consisted of the construction of 21 models to predict the temperatures of 7 heating zones. The prediction accuracy of each model has been tested for three different horizons, as one of the goals of creating these models is to employ them along with advanced control and optimization techniques. The results show a grandiose ability of these models to accurately predict temperatures for short, medium, and long forecast horizons.

The importance of quality and treatment of the database was fundamental to obtaining positive results because each model was properly analyzed to provide the expected robustness, as well as the reduction of the amount of data, which has a total influence on the computational processing capacity. The results show a grandiose ability of these models to accurately predict temperatures for short, medium, and long forecast horizons. It is important to highlight that the models must be trained for an optimal performance one step ahead, that is, in the short term, because as we advance to predictions of medium or long horizons, we have the degradation of the prediction.

RECOGNITIONS

We thank the Federal Institute of Espírito Santo, Propecaut program, ArcelorMittal Tubarão, for the partnership and incentive to technological and human development and the Cooperation Term 133/2021 CAPES/FAPES for the financial support.

REFERENCES

Trinks, w.; mawhinney, m.; shannon, r.; reed, r.; garvey, j. Industrial furnaces. 6. Ed. New york: john wiley & sons, 2004.

Hu, yukun et al. Nonlinear dynamic simulation and control of large-scale reheating furnace operations using a zone method based model. Applied thermal engineering, v. 135, p. 41-53, 2018.

Vanderschueren, d. 2020 technology and operation of a hot rolling mill. In: iop conference series: materials science and engineering. Iop publishing, 2020. P. 012020.

Liao, ying-xin; she, jin-hua; wu, min. Integrated hybrid-pso and fuzzy-nn decoupling control for temperature of reheating furnace. Ieee transactions on industrial electronics, v. 56, n. 7, p. 2704-2714, 2009.

Lu, biao et al. A novelty data mining approach for multi-influence factors on billet gas consumption in reheating furnace. Case studies in thermal engineering, v. 26, p. 101080, 2021.

Chen, chien-jung; chou, fu-i.; chou, jyh-horng. Temperature prediction for reheating furnace by gated recurrent unit approach. Ieee access, v. 10, p. 33362-33369, 2022.

Zhai, naiju; zhou, xiaofeng. Temperature prediction of heating furnace based on deep transfer learning. Sensors, v. 20, n. 17, p. 4676, 2020.

Bao, qingfeng et al. Multivariate linear-regression variable parameter spatio-temporal zoning model for temperature prediction in steel rolling reheating furnace. Journal of process control, v. 123, p. 108-122, 2023.

Zhang, pinggai et al. Furnace temperature prediction using optimized kernel extreme learning machine. In: 2021 40th chinese control conference (ccc). Ieee, 2021. P. 2711-2715.

Hochreiter, sepp; schmidhuber, jürgen. Long short-term memory. Neural computation, v. 9, n. 8, p. 1735-1780, 1997.

Bengio, yoshua; simard, patrice; frasconi, paolo. Learning long-term dependencies with gradient descent is difficult. Ieee transactions on neural networks, v. 5, n. 2, p. 157-166, 1994.

Chung, junyoung et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. Arxiv preprint arxiv:1412.3555, 2014.

K. Cho, b. Van merrienboer, d. Bahdanau, and y. Bengio. On the properties of neural machine translation: encoder-decoder approaches. Arxiv preprint arxiv:1409.1259, 2014.

Kuan, lu et al. Short-term electricity load forecasting method based on multilayered self-normalizing gru network. In: 2017 ieee conference on energy internet and energy system integration (ei2). Ieee, 2017. P. 1-5.

Lea, colin et al. Temporal convolutional networks for action segmentation and detection. In: proceedings of the ieee conference on computer vision and pattern recognition. 2017. P. 156-165.

Xu, yuanhao et al. Application of temporal convolutional network for flood forecasting. Hydrology research, v. 52, n. 6, p. 1455-1468, 2021.
Bai, shaojie; kolter, j. Zico; koltun, vladlen. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. Arxiv preprint arxiv:1803.01271, 2018.
Oord, aaron van den et al. Wavenet: a generative model for raw audio. Arxiv preprint arxiv:1609.03499, 2016.

Hashempour, s. Et al. Continuous scoring of depression from eeg signals via a hybrid of convolutional neural networks. Ieee transactions on neural systems and rehabilitation engineering, v. 30, p. 176-183, 2022.

Yan, jining et al. Temporal convolutional networks for the advance prediction of enso. Scientific reports, v. 10, n. 1, p. 1-15, 2020.

Pollhammer, werner et al. Modeling of a walking beam furnace using cfd–methods. Energy procedia, v. 120, p. 477-483, 2017.

Cadena-ramírez a, favela-contreras a, dieck-assad g. Modeling and simulation of furnace pulse firing improvements using fuzzy control. Simulation. 2017;93(6):477-487. Doi:10.1177/0037549717692418

Jo, jun-mo. Effectiveness of normalization pre-processing of big data to the machine learning performance. The journal of the korea institute of electronic communication sciences, v. 14, n. 3, p. 547-552, 2019.

Shi, jimeng; jain, mahek; narasimhan, giri. Time series forecasting (tsf) using various deep learning models. Arxiv preprint arxiv:2204.11115, 2022.

Kingma, diederik p.; ba, jimmy. Adam: a method for stochastic optimization. Arxiv preprint arxiv:1412.6980, 2014.

Kaplunovich, alex; yesha, yelena. Automatic tuning of hyperparameters for neural networks in serverless cloud. In: 2020 ieee international conference on big data (big data). Ieee, 2020. P. 2751-2756.

Chai, tianfeng; draxler, roland r. Root mean square error (rmse) or mean absolute error (mae)?– arguments against avoiding rmse in the literature. Geoscientific model development, v. 7, n. 3, p. 1247-1250, 2014.